

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

Field of exception situation recovery invention of data processing system This invention relates to the exception situation method of recovery and equipment of data processing system. In especially this invention operating the data processing system controlled by the sequence of the instruction which forms a program (a) A step including the data transfer of the 1st type depending on the machine condition that performed said program while specifying the number of the checkpoints of said program, and said conversion was specified with the data transfer in said system, (b) The step which answers the exception situation produced during activation of this step (a), and interrupts said program execution, (c) It is related with the data-processing-system actuation approach equipped with the step which reruns said program from the latest checkpoint after solution of said interruption.

Moreover, this invention is data processing system. A means to generate the sequence of the instruction which forms a program on the occasion of activation by this system, A means to identify the checkpoint in said program While performing said program, data transfer of the 1st type is performed. The 1st data path for which each of such data transfer depends on the machine condition that said system was specified, A means to answer the exception situation and to interrupt said program execution In data processing system equipped with a means to rerun a program from the recently checkpoint after solution of said exception situation Said system A means to connect with said data path and to record each of said 1st type of data transfer from a checkpoint recently [said] While interrupting said account data path of rerun Nakamae It is related with the data processing system characterized by having further a means to reproduce the recorded data transfer instead of actual data transfer synchronizing with said rerun.

Such an approach and a system are known from 1469-1514 pages of "Checkpoint Repair for High-Performance Out-of-Order Execution Machines" by W.-M.W.Hwu and Y.N.Patt of IEEE Transactions on Computers published in December, 1987, Vol.C-36, and No.12.

Explanation of a related technique Data processing system is various "exception situations" produced while performing a program.

It needs to be alike and it is necessary to harmonize and cope with it. The exception situation contains overflow and the page fault of interruption and count. (It was generated by the external asynchronous event) If actuation of these interruption is delayed, it can process by **, until a system reaches the condition which can save interruption after interruption is processed, and can be recovered of having been specified good. This is 68000. It is carried out with a microprocessor family and interruption is processed in this case only on a micro structure boundary. To it, it can be considered that overflow of count is serious to activation of a part of program, consequently the relation of it is lost with a machine condition.

Other criteria of the exception situation are generated between the usual program executions, and control is given to hardware from software in this case. The main examples as which effective processing is required are the "page faults" of a virtual storage system, and become accessible [an addressable predetermined part] at the time of predetermined in an instant in this case. Other examples are TLB of a

system in case the trap and storage management unit (MMU) which emulate the op code which is not performed do not have table walk hardware. The processing in the special case in count of the Myst lap and the point floating [IEEE] is included. This type of exception situation is processed in an instant, no matter the condition of a machine may be what thing, and quick and the utility of the bulk memory which needs to process so that it may have dependability, namely, has a virtual addressing are lost. For example, MC68020 After processing the exception situation, it is necessary to make it recover, while carrying out the stack of the condition of the relation of a processor then and saving. Since a machine has the working decision nature, it is made for a machine to resume actuation to accuracy to all the practical objects by making the condition that it turns out that it is effective recover a machine, so that the exception situation may never occur.

The processor of the higher engine performance is in the inclination which becomes denser pipeline arrangement. Since an additional hardware data path is required, the condition of such a machine becomes very large and most conditions exist in an expensive pipeline stage to access. Therefore, it comes to specify the "checkpoint" in the case of being special in a program actually. If it arrives at a checkpoint while performing, as long as it relates to future actuation, backup copy forms a machine condition. After processing the exception situation, it is necessary to restore the condition of a processor to the thing of the latest checkpoint, and to resume activation from a checkpoint.

The exception situation recovery using a checkpoint is indicated by 1469-1514 pages of "Checkpoint Repair for High-Performance Out-of-Order Execution Machines" by W.-M.W.Hwu and Y.N.Patt of IEEE Transactions on Computers published in December, 1987, Vol.C-36, and No.12. Hwu And Patt maintains the backup copy of a small register file, and explains the approach and equipment which restore a machine to the condition of a checkpoint. Backup copy is not practical to the mass machine condition which contains the full capacity of main memory in many cases. Therefore, Hwu And patt saves the value overwritten by the data transfer to the memory produced from a checkpoint recently, and in order to restore a condition before resuming activation from a checkpoint, it explains how to write in this value. Although the saved value occupies a very small tooth space, the rewrite of these values wastes time amount and has a possibility that the whole processing system performance may fall.

A checkpoint retry system is the Europe patent specification 0348652nd. It is indicated by the number. It is held at a specific buffer until the instruction whose value written in cache memory specifies a checkpoint is completed, and a change of state is delayed in this system until it turns out by this that the exception situation has not arisen.

The type of the processor for which it depends on parallel execution and dense pipeline arrangement substantially is a "very long instruction word" (VLIW) type processor. "Ideal VLIW" processor and TRACE A known practical use VLIW processor as a processor 1987 year 10 month 5-8 On a day Palo Alto of the California week () [Palo] Architectural Support of Alto for Programming Languages and Operating Systems Proceedings of (ASPLOS II) of the Second International 180-192 of Conference R.P.Colwell of a page etc. -- it is indicated by the paper of the title of 'A VLIW Architecture for a Trace Scheduling Compiler' to twist. exception situation recovery -- being related -- Colwell etc. -- once the exception situation is solved, the "history queue device" in which interrupted memory actuation is performed again will be explained vaguely.

The video signal processor which has VLIW architecture is indicated by the above-mentioned Europe patent specification title No. (PHN 12.376) 325,310 which carries out relation. For the specific application of video signal processing, the small cyclic program which used the physical memory is used, consequently the problem of the exception situation which can be restored does not arise. The same configuration of a more common processor will be 3-5 in October, 1988. Computer Design of a day 'CREATE-LIFE by this invention person of Proc.IEEE International Conference : A Design System for High Performance VLSI circuits' and the Europe patent specification title 479,390 It is indicated by the number (PHA 1209). Since it is desirable to require mass virtual memory when using this structure for a general processing application, the problem of the exception situation arises. It is checked about loss of a circuit dimension and/or performance that the disadvantageous profit using a known recovery system becomes severe especially by the VLIW processor.

Europe patent specification 479,390 Other reference explaining the VLIW processor indicated by the number has not made reference in the approach of the exception situation recovery fitted to it (responding to a case).

The approach these systems "cannot delete a control pipeline" easily, especially other problems relevant to a known exception situation recovery system decrease a circuit dimension by the other approach to predetermined performance level in a VLIW processor is needed.

Epitome of invention The object of this invention is disadvantageous and/of performance. Or it is offering other exception situation methods of recovery which decrease a circuit dimension. The object of this invention is offering the suitable exception situation method of recovery using a dense pipeline processing system like a VLIW processor especially.

In operating the data processing system controlled by the sequence of the instruction which forms a program, this invention is equipped with the above-mentioned step (a), (b), and (c), and said step (a) offers how said step (c) synchronized with said rerun instead of actual data transfer including playback of the recorded data recently [said] including record of said 1st type of each data transfer from a checkpoint.

Instead of restoring the machine condition before the rerun in a checkpoint, the approach by this invention disregards the present machine condition, and constitutes the recorded value which is acquired from the exact machine condition before exception situation generating. Hwu And the small circuit dimension of the method of patt is maintained, and the inconvenience of performance including restoration of an exact machine condition is removed simultaneously. A known approach restores all the values overwritten from the checkpoint irrespective of whether it has the effect of arbitration on rerun. The approach by this invention memorizes and reproduces only the value which was used for original activation at accuracy, therefore is required of rerun to it.

"programmer which includes the content of the pipeline stage in programming like a processor and which was limited dramatically -- visible --" condition is specified. Under the present circumstances, a checkpoint specifies the point of the program relevant to actuation of the future [condition / this / visible]. a single string of each of the fragmentation which has arranged by dividing this into the fragmentation which carried out checkpoint assignment of the program, and carried out checkpoint assignment in this way is working -- it has only one pre DESSA which may exist. If it puts in another way, a control flow will be branched suitably in a checkpoint. A means in the specification which cannot avoid branching to be able to carry out the schedule of the program so that only a visible condition may relate to a junction (a pipeline is empty), namely, to satisfy an exact condition is established.

Said step (c) may also include the restart of usual activation of said program while recording said 1st type of data transfer according to said step (a), if said rerun reaches the point that said activation was interrupted for said step (b). By this approach, the exception situation of other arbitration following the same checkpoint can be coped with automatically.

When required, the step required of solving the exception situation is a known approach, and it depends for it on this contractor containing specific hardware and/or a specific software element at the property of the exception situation. These steps do not form a part of this invention.

As a result of interruption of activation of said step (b), this transmitted control of said system to the exception situation manipulation routine from said program, was delayed in the transfer of said control, and the example which enables it to complete those activation explained it with the instruction generated before said exception situation. It can carry out as [affect / the specified machine condition / forbid simultaneously the actuation started by the delay before a transfer of the control to an exception situation manipulation routine, and / these actuation]. In case the usual exception situation resumes from the point of interruption, these means can be used if needed so that a machine condition may become exact. Actuation can be forbidden using the "guard" facility of a LIFE type VLIW processor, without using hardware especially. By using this facility (or what is equivalent to this), it says [it / that this arises] with the conventional exception situation recovery system and is satisfactory, and control of pipeline arrangement can also be deleted.

This invention offers the suitable data processing system and the equipment for actuation according to

the approach by above-mentioned this invention.

Easy explanation of a drawing The gestalt of operation of this invention is explained to a detail with reference to a drawing.

Drawing 1 shows the linearity gestalt of the processor by this invention.

Drawing 2 shows a format of the very long instruction word of the equipment of drawing 1.

Drawing 3 shows a part of imitation multiport memory of the equipment of drawing 1.

Drawing 4 shows actuation of the program fragmentation of the 1st example.

Drawing 5 shows the scheduling of the program fragmentation of the 1st example to the equipment of drawing 1 including the problem of a collision.

Drawing 6 shows some processors including exception situation detection and a recovery circuit.

Drawing 7 shows record/regenerative circuit of a processor more to a detail.

Drawing 8 shows the scheduling of the program fragmentation of the 2nd example.

Drawing 9 shows actuation of the exception situation recovery circuit under 2nd-example activation of program fragmentation.

Drawing 10 shows other examples of record/regenerative circuit of a processor to a detail.

Scheme of a processor The processor of drawing 1 is equipped with the imitation multiport store circuit 30 connected to five functional units which contain the branching unit BR in the constant unit CO, count and Logical unit AL, the register unit RG, and a data memory unit DM list. The specific register SP which has the storage unit 26, a multiplexer 27, and the multiplexer 29 of relation in a store circuit 30 is illustrated in linearity. The principle of such actuation of a store circuit and a processor is indicated by the Europe patent specification No. (PHA1209) 479390, SURABEMBAGU and a love loss could refer it, and both sides already explained it. These principles did not form a part of this invention, but since it was simple, those publications were omitted.

Sequencer SEQ And instruction generating pipeline IIP The global controller 41 which it has is formed. Sequencer SEQ Instruction generating pipeline IIP Program counter WORD PC supplied is generated in each cycle. Instruction generating pipeline IIP Being able to consider as an OFU chip, this contains the program memory by which the very long instruction word which forms an executive program is encoded. Instruction-after delay of predetermined number of cycles generating pipeline IIP The very long instruction word IW memorized to the address PC of program memory is supplied to the instruction register 46 on chip.

The instruction word IW of a register 46 is destination control field DEC. Functional units CO, AL, RG, and DM according to individual And two or more fields COC, ALC, RGC, DMC, and BRC which control BR, respectively It has another bit CP which marks a checkpoint. although it is an easy example for the explained pro SESSAHA emergency -- the total of the bit of instruction word IW -- 163 it is.

Drawing 2 shows the field of instruction word IW more to a detail, and these fields have the number of bits of each field attached under the field. The object of these various fields is making it clear from the configuration of a processor, and detailed explanation of the following of operation. The object of the checkpoint marking bit CP is explaining to a detail below with reference to drawing 6 -10 of a drawing. The constant unit CO does not have the operand input section, and has the output section (CO.O) as a result of 32 bits linked to the input section of a store circuit 30. The constant unit CO is 32 bit field COC of instruction word IW (drawing 2) actually. It has the direct continuation between the output sections of the constant unit CO. Thereby, the constant unit CO can supply constant value to other units from within the limits of instruction word through a store circuit 30.

Count and Logical unit AL have the guard value input section (AL.G) of 1 bit and the two 32-bit variable input sections (right and left) (AL.L and AL.R) linked to each memory unit in a store circuit 30. Unit AL has the 32-bit output section (AL.O) which made reverse connection in a store circuit 30, and is the 16 bit field ALC of instruction word IW. It is controlled.

Drawing 3 shows a part of imitation multiport store circuit 30 relevant to count and Logical unit AL to a detail. In a store circuit 30, they are operand AL.G and AL.L about the storage units (RAM) 48, 50, and 52 according to individual. And it is made to relate to AL.R, respectively.

Furthermore, if it contrasts with other usual processors, the write-in enable signal according to

individual will not be supplied to the storage unit in the equipment of drawing 1 -9. Instead, in case it does not ask for at least one writing of as opposed to [exist namely] the storage unit for the storage location in which at least one writing is impossible, it can address. If it assumes that it is already known when the value of a write-in enabling bit is scheduling, in case a result will be assigned to the location of a specific storage unit, the width of face of instruction word IW is decreased when these enabling bit does not exist.

The storage unit 48 to guard bit AL.G has eight 1 bit positions including an explained location in which one writing is impossible for memorizing a Boolean value (TRUE/FALSE bit). The triplet read-out address to the storage unit 48 is control field ALC of instruction word IW by the instruction register 46. It is supplied from 3 bit-field AL.G.RA (drawing 2), and the 1-bit read-out port 54 of a storage unit supplies guard bit AL.G to Unit AL from the addressed storage location.

A multiplexer 56 has two or more 1-bit input sections linked to the 32-bit minimum important bit (lsb) connected to the output sections (CO.I, AL.O, etc.) as a result of various functional units. About the input section of an addition of a multiplexer, it is lsb of the additional 32-bit bus line 58. It connects, and this conveys output SP.O from the delay element of the gestalt of the specific register SP, as a result of being delayed. When it is used preparing the delay element which solves the collision of this type of VLIW, it is the above-mentioned Europe patent specification 479,390th. The technical problem of a number (PHA1209) is formed. It is lsb to a guard bit. When it is used, it comes to have compatibility with the predetermined programming convention which includes programming of C, for example. In other situations, the maximum important bit (msb) or other bits can be chosen more as convenience. A multiplexer 56 has the 1-bit output section 60 linked to the write-in port of the storage unit 48. A multiplexer 56 is destination control field DEC of instruction word IW. Selection field AL.G.SE of a triplet It is controlled. The write-in address of the triplet to the write-in port of the storage unit 48 is destination control field DEC. Adjoining 3 bit-field AL.G.WA It is given.

Similarly, the 32-bit storage units 50 (as opposed to AL.L) and (as opposed to AL.R) 52 have the 32-bit multiplexers 62 and 64 of relation, respectively.

Each has the location including the location in which one writing is impossible of 16 pieces. Destination control field DEC Two triplet selection field AL.L.SE which controls multiplexers 62 and 64, respectively And AL.R.SE Two 4 bit-field AL.L.WA(s) which supply a 4-bit write-in port address to each of the storage units 50 and 52 And AL.R.WA It contains. Control field ALC Two 4 bit-field AL.L.RA And AL.R.RA The read-out address to the read-out port of the storage units 50 (as opposed to AL.L) and (as opposed to AL.R) 52 is generated.

op code AL.OP to count and Logical unit AL Control field ALC to Unit AL It is given by five bit fields. It sets in each cycle and is op code AL.OP. In order to generate result AL.O, it is determined [which] to argument AL.L and AL.R whether activation (addition, subtraction, a comparison, AND, OR, etc.) of operation should be carried out.

If it returns to drawing 1 , the register unit RG will offer the storage on chip to 16 program variables whose each is 32 bits and which are used frequently. Unit RG has the storage unit of the relation in the store circuit 30 to guard bit RG.G and 32-bit data input RG.I. These storage unit has only four locations where each includes the location in which one writing is impossible. Data output (RG.O) is connected to the input section of the store circuit 30 which forms each bus line (refer to RG.O and drawing 3).

Destination control field DEC Selection field RG.G.SE of two triplets which control the multiplexer of relation And RG.I.SE Two 2-bit write-in address-field RG.G.WA(s) chosen among four locations of each storage unit And RG.I.WA It contains.

9-bit register unit control field RGB Inside, they are two 2 bit-address RG.G.RA. And RG.I.RA It prepares in order to choose from each storage unit the location of the value of value RG.G which should be carried out reading appearance, and RG.I. 5-bit op code field RG.OP It has 1 bit chosen between read-out actuation of REJITA, and write-in actuation of a register, and 4 bits chosen among 16 registers in Unit RG. Therefore, the location of the variable in the register unit RG is fixed by the compiler.

The data storage unit DM includes a lot of storage to a variable at least notionally. Actually, Unit DM is equipped with the interface over a bank of read/write memory on chip so that it may explain to a detail

later with reference to drawing 6 . In the imitation multiport store circuit 30, the storage unit according to individual is prepared to guard bit DM.G and 32-bit address word DM.A and 32-bit close KADETA WORD DM.I. These storage unit has the location including the location which cannot write in each of 256 pieces. The data storage unit DM has the 32-bit data output (DM.O) connected to the input section of the storage unit 30 which forms each bus line (also see DM.O and drawing 3). Destination control field DEC Three triplet selection field DM.G.SE and DM.A.SE which control the multiplexer in a store circuit 30 And DM.I.SE Each 256 of the storage unit of relation Three 8-bit write-in address-field DM.G.WA(s) and DM.A.WA which are chosen between the locations of an individual And DM.I.WA It contains.

25-bit data storage unit control field DMC Inside, they are three 8-bit read-out address DM.G.RA and DM.A.RA. And DM.I.RA Value DM.G which should be carried out reading appearance from each of those storage units, and DM.A And it prepares in order to choose the location of DM.I. 1-bit op code field DM.OP 1 bit chosen between read-out actuation of storage and write-in actuation of storage is included. Address word DM.A specifies the address to the writing of data word DM.I, or read-out of data word DM.O. Thereby, dynamic count of 32-bit sufficient address to address the location of 232 can be performed among program execution.

Sequencer SEQ which should be corrected so that it may branch during program execution with the branching unit BR The sequence of generated instruction address PC is permissible. Unit BR has the storage unit of the relation to guard bit BR.G and 32-bit destination address BR.A in a store circuit 30. These storage unit has four locations including the location which cannot write in each. Destination control field DEC Two triplet selection field BR.G.SE which controls the multiplexer of relation And BR.A.SE Two 2 bit write-in address-field BR.G.WA(s) chosen among four locations of each storage unit And BR.A.WA It contains.

4-bit branching unit control field BRC Inside, they are two 2-bit read-out address BR.G.RA. And BR.A.RA It prepares in order to choose from each storage unit the location of value BR.G which should be carried out reading appearance, and BR.A. The branching unit BR is Sequencer SEQ about guard bit BR.G and destination address BR.A. It is made to pass simply. op code field BR.OR It is not required. Sequencer SEQ The Boolean value of guard bit BR.G is tested inside. When BR.G is truth, branching should be performed, and it is Sequencer SEQ. The next value of PC equal to destination address BR.A which received from the store circuit 30 is created. It is Sequencer SEQ when BR.G is a false. The increment of the value of PC is only carried out to PC+1, and the next instruction of the memorized program is addressed.

Therefore, what (the truth or the false of lsb) logic result AL.O is taught to the storage unit to BR.G for can perform facultative branching. (Or it is always carried out, it is never carried out) A fixed result teach [branching which is not facultative / a storage unit / as opposed to BR.G for CO.O], i.e., when a storage unit always arranges truth or the location which always becomes false and which cannot be written in, it can perform. Since at least one location of each storage unit always cannot write in the addition of this latter as already explained, it is convenient especially in the gestalt of operation of this invention. (Branching is never performed) Guard bit BR.G=FALSE is often required, when branching actuation should be started in a predetermined cycle.

Sequencer SEQ Bus line (refer to PPC and drawing 3) False program enumerated data PPC memorized by the input section of the store circuit 30 to form It is made to generate. PPC other functional units containing the branching unit BR -- using it -- a program -- counting - correlation addressing can be performed. For this reason, it is the value PPC of a bus line to a storage unit about read-out delay. The number of cycles and PPC of an instruction to read It can specify as a difference between the numbers of cycles of an instruction which have PC equal to a value.

Cycle i and instruction word IW (i+b) which branching delay of working and b cycle makes generate branch instruction 1st cycle i+b performed by branch instruction It generates in between. The change-of-state actuation which can teach the same Boolean value of the conditions of a decision instruction to guard bit input section AL.G, RG.G, DM.G, and BR.G, and includes another branching and by which the schedule was carried out to the various functional units under branching delay can be used for the

mediation cycle forbidden facultative. By getting to know the possibility of branching, within branching delay of the facultative decision instruction which may happen, the schedule of the effective actuation can be carried out with a compiler, and in case branch condition suits, when simultaneously few, an exact condition is satisfied.

In order to start the actuation which is not guarded with the most sufficient convenience, suitable read-out address .R.A is answered in the location which cannot write in each guard bit storage unit, and it arranges so that the read-out port may be made to generate truth. It can also arrange so that the value of which the location which cannot write in a 32-bit operand storage unit is required frequently [of a certain known], for example, zero, may be generated.

Only the register of off chip memory and the register unit RG and the content of the program counter PC are considered to be the parts of the programmer visible condition of the processor of this gestalt. The IMITETAYON multiport store circuit 30 can be strictly used with the compiler as a buffer to a temporary medium variable. Consequently, a functional unit like the constant unit CO is not needed for a guard. Even if count and Logical unit AL itself cannot change the programmer visible condition of equipment, the guard who avoids an exception situation like overflow of the count produced in false to Unit AL is prepared.

In addition to branching delay of the branching unit BR, a scheduling compiler needs to have the so-called information of "latency" of each functional unit. While generating the number and the results (AL.O etc.) of actuation of the cycle which starts actuation for the latency of a functional unit, it is destination control field DEC. It specifies as a difference between the number of cycles which teaches the result to the location of one or more requests of the imitation multiport store circuit 30. About the already explained branching delay, the branching unit BR is the latency and Sequencer SEQ of a cause. And off-chip instruction generating pipeline IIP The sum with the latency of a cause can be considered. Example of scheduling For the following examples, the latency to a functional unit is assumed as follows.

Unit CO-0 cycle latency Unit AL-1 cycle latency Unit RG-1 cycle latency Unit DM-two-cycle latency Unit BR-5 cycle branching delay The following list 1 is equipped with the fragmentation of a high-level language source code. Variables p and q show two records in memory. The objects of a program are two things to copy to the field of corresponding p record from the field of q record of two values of 32 bits.

List 1 $p^{\wedge} \text{ field } 1 := q^{\wedge} \text{ field } 1;$

$p^{\wedge} \text{ field } 2 := q^{\wedge} \text{ field } 2;$

Drawing 4 illustrates the configuration which may have data which can be used for executing a high-level instruction of a list 1. High-speed register file REG which has the location of 16 pieces which attached the number of 0 to 15 for storage of a program variable And (off chip) bigger bank MEM than that of read/write memory It carries out. About the local base address LB, it is a register file REG. A location 0 is made to memorize. Address LB is a memory bank MEM. Initiation of the table of an adjustable value is shown and two record pointers p and q are location LB+12 in this case. And it is recorded on LB+8, respectively. About each variables p and q, it is a memory bank MEM. It considers as the pointer to initiation of corresponding record arranged except. Offset of zero is made to memorize the 1st field of each record from the address to which it was specified by the pointer variable of relation. Offset of 4 is made to memorize the 2nd field of each record from the pointer variable of relation. therefore, the value $q^{\wedge} \text{ field } 2$ -- the address -- $q+4$ it is -- it can read from a location.

A list 2 shows the pseudo code fragmentation which attached T1 which makes it generate with the compiler which executes a high-level instruction of a list 1 by the VLIW processor, when the data configuration explained with reference to drawing 4 is assumed. Each line of a list 2 is single basic actuation Op001 -Op014 which should be carried out a schedule to activation by the suitable functional unit. It specifies. The comment (*, --, *) of the end of each line makes reference in the types (AL, RG, CO, etc.) of the functional unit demanded first, and explains the result (output) of the actuation after that.

(for example) Actuation Op003 Reference number 001 And 002 Op003 The receiving operand is the result of Op001, and Op002, respectively. It expresses that it is a result. Therefore, it is actuation Op001

about the effect of actuation Op003. A result and actuation Op002 It adds to a result and is actuation Op009 simultaneously. By effect, it is Op008. It is a result Op004 of off chip data memory It should write in the location addressed by the result. the list 2 which will perform the function specified by the list 1 if explanation of above-mentioned drawing 4 is combined with this information -- each -- of operation Op001 -Op014 The object becomes obvious.

List 2T1:Op001 rgreg (0); (* RG and local base address *)

Op002 constant (12); (offset value * to * CO and p)

Op003 aplus 001, 002; (address [of * AL and p] *)

Op004 read 003; (* DM,p *)

Op005 constant (8) (offset value * to * CO and q)

Op006 aplus 001, 005; (address [of * AL and q] *)

Op007 read 006; (* DM,q *)

Op008 read 007; (* DM, q[^]. field 1 *)

Op009 write 004 008; (* DM, p[^]. field 1:=q[^]. field 1 *)

Op010 constant (4); (offset * to * CO and the field 2)

Op011 aplus 004, 010; (address [of * AL and the p[^]. field 2] *)

Op012 aplus 007, 010; (address [of * AL and the q[^]. field 2] *)

Op013 read 012; (* DM, q[^]. field 2 *)

Op014 write 011, 013; (* DM, p[^]. field 2:=q[^]. field 2 *)

GOTO EXIT The sequential list of the pseudo code fragmentation T1 is carried out, and, in a suitable case for at least one functional unit to perform each actuation, it turns out that this fragmentation is not specified as the specific gestalt of the arbitration of a VLIW processor. Once a compiler once reaches collection of such fragmentation that constitutes an application program and a actual equipment gestalt serves as known, it is the task of a scheduler to map each actuation of fragmentation in the specific functional unit of the equipment of a specific cycle.

A scheduler operates by constraint of the lot specified by the configuration of a specific VLIW processor. These constraint mainly contains the number of functional units and type which can be used by the target VLIW processor, and the latency of each functional unit. Other constraint contains the number of the locations of each storage unit of an imitation multiport memory.

These locations are dynamically assigned if needed by the scheduler. For the problem of easy scheduling expressed by fragmentation T1, the number of the locations of each storage unit of the VLIW processor of drawing 1 -4 appears in extent which does not have to consider a limit of the capacity of a storage unit as constraint of this description enough, and a certain thing is understood.

Since it is not necessary to carry out a schedule until it makes the latency of the actuation of everything but all by actuation depending on the result of other actuation complete, the latency of each actuation is important. Therefore, actuation Op009 They are a two cycle and Op008 at least after Op (DM latency is 2)004. It is necessary to carry out a two-cycle schedule at least later. Similarly, it is actuation Op011.

Although a schedule can be carried out in the same cycle as Op (CO latency is zero)010, it is necessary to consider as the two-cycle back at least than Op (DM latency is 2)004. This assumes that the writing and read-out of a storage unit location can be performed simultaneously. When that is not right, there is a possibility of needing another latency of 1 cycle for actuation of each type.

Drawing 5 shows three different schedules (I, II, and III) which each maps in the specific VLIW processor which explained the pseudo code fragmentation T1 with reference to drawing 1 -4. Each line of each table expresses one very long instruction word IW, therefore one machine cycle. The cycle number which attached Cy to the head is given to the leftmost column. The following five columns are five control fields COC, ALC, RGC, and DMC to five functional units (CO of drawing 1 etc.). And BRC It expresses. Each control field start actuation with each functional unit. Therefore, at Schedule I, it is Field COC in a cycle 0. It orders to a constant unit (CO of drawing 1) to start Op005, consequently appears in the output section of the unit CO whose result CO.O=8 are the same (CO latency is zero) cycle. Similarly, it is Field RGC in a cycle 0 about the read-out actuation Op001 of a register. It starts. The value LB of a local base address follows and is RG.O in a cycle (RG latency is 1) 1.

Destination control field DEC under each instruction Selection field DEC. -- .-.SE It expresses to the right of functional unit control field. These fields are various of operation Op001 -Op(s)014. A result is taught to each storage unit so that such a result can be used.

A scheduler is specified destination control field DEC. The latency of various functional units is taken into consideration. therefore, the field RGC of a cycle 0 from -- started register read-out actuation Op001 a result -- RG.O -- the next instruction and field DEC.AL.L.SE of a cycle 1 It is taught to count and the left-hand side operand storage unit of Logical unit AL. This is expressed with the code RG of the AL.L train of the cycle 1 of Schedule I. the same -- the memory read-out actuation Op007 -- the field DMC of a cycle 2 from -- starting -- simultaneous -- the result of the actuation -- DM.O -- a cycle (DM latency is 2) 4 -- field DEC.DM.A.SE It teaches the address storage unit of the data storage unit DM by setting up 'DM'.

A scheduler is the actuation Nw015 of itself. And Nw016 It adds and the jump by the following fragmentation is performed ('GOTO EXIT' of a list 2). Nw015 The constant unit CO is used in a cycle 1, and it is selection field DEC.BR.A.SE in a cycle 1. By setting it as 'CO' prescribes the target address to branching taught to the address storage unit of the branching unit BR. Nw016 It is the branching-control field BRC in a cycle 1. It uses and unconditional (BR.G=1) branching to the target address is started. A transfer of the control to the target address stops arising to a cycle 8 after fragmentation T1 completion by long branching delay of 5 cycles. If it puts in another way, branching actuation is shown into a schedule and can carry out the schedule of the effective actuation to the activation under branching delay. The schedule only of the abbreviation 1/3 of actuation of the maximum number which may happen is actually carried out to the schedule I of drawing 5 . It is because this cause is the example of the fragmentation of very small size, and since extent of coincidence becomes high as a flag mate becomes long, if the schedule of the longer fragmentation is carried out, it can be understood to this contractor that the schedule of the effective high actuation of a rate can be carried out.

It is conquered with the gestalt of the operation explained to longer fragmentation by utilization of guard bit .G of a Boolean value which can use that facultative branching actuation of many propers can arrange most facultative branching so that it may have a branching probability close to 1 or 0. For example, branching it can be predicted that produces the loop formation which should be executed 100 times in the probability of 0.99 or 0.01 is meant.

The schedule I of drawing 5 is performed by the VLIW machine which has a true multiport memory, and it is destination control field DEC. It ignored thoroughly. However, Schedule I is of operation Op001 -Op014 to the processor of drawing 1 -4, including [therefore] the collision of access to the storage unit of an imitation multiport memory. It is not an effective map. Especially, in the cycle 5 of Schedule I, it is Op004. Result DM.O is sent to count, the left-hand side operand storage unit (AL.L) of Logical unit AL, and the address storage unit (DM.A) of the data storage unit DM. It sets in the same cycle 4 and is Op012. Result AL.O is also sent to the address storage unit (DM.A) of the data storage unit DM. It sets in the cycle 5 of Schedule I, and this collision is field DEC.DM.A.SE. It is marked by Code XX.

When a delay element (the specific register SP) does not exist, a scheduler is forced so that the re-schedule of these actuation may be carried out, in case a collision is discovered, until constraint of processor hardware is satisfied. In this example, this serves as the schedule II of drawing 5 .

It sets on Schedule II and is Op004. 1 cycle delay is carried out and it starts in a cycle 4. However, actuation Op009 and Op011 And Op014 Op004 Depending on a result therefore, at least one cycle also of these actuation is also delayed. It sets on Schedule I and these actuation is other actuation Op012, Op008, and Op013. And Op014 It is necessary to carry out the re-schedule also of these actuation by overlap and hardware constraint mutually.

Although the process of this re-schedule, i.e., "back track actuation", can be made to complete so that hardware constraint may always be satisfied, nine cycles of fragmentation T1 by which the re-schedule was carried out are occupied instead of 8 in this case. This means taking this long part of an application program compared with the case of the ideal VLIW machine which has the same functional unit.

Furthermore, back track actuation means the compiler including solution of the new collision generated

in case the first collision is solved in an additional work piece. It becomes serious especially by the big machine by which a collision almost generates the problem of the effect of the increasing compiler in a cycle at each time. It is clear that existence of a collision is generally undetectable until the schedule of two or more instructions is carried out exceeding the cycle which the collision generated. By 10 or the big machine of the functional unit beyond it, this requires the re-scheduling of dozens of times or hundreds of actuation to which the new collision produced as a frequent thing during re-scheduling may take place.

The schedule III of drawing 5 is the simultaneous continuation application 594,534. The analysis of the example of the collision without back track actuation using the delay element of the gestalt of the specific register SP which was explained in detail by the number is shown. Destination selection field DEC Selection field DEC.SP.SE It is shown in the right-hand side of Schedule III. Of operation Op001 - Op014 And Nw015 -Nw016 It turns out that it is almost the same as that of the thing of Schedule I (precise multiport memory). However, the collision of the cycle 5 of a between is Op004 two results which goes to a DM.A storage unit in Schedule III. Teach result DM.O to a specific register, i.e., field DEC.SP.SE. Destination control field DEC By setting it as 'DM', it is solvable. Field DEC.DM.A.SE In this case, it is set as 'SP' in a cycle 6, and is Op009 of a cycle 6. It is Op004 when reading appearance should be carried out with the data storage unit DM. The result of having been delayed is taught to a DM.A storage unit.

Op004 The result of having been delayed collides with the result of Op011 in a cycle 6 in this case, and, as for these results, both sides go to a DM.A storage unit. Therefore, it sets on Schedule III and is Op011. Initiation is delayed to a cycle 6, without having continuous effect on actuation of other arbitration. It sets in a cycle 5 and other solutions to this 2nd collision are Op011. It leaves as it is, it sets in a cycle 6 simultaneously, and is field DEC.SP.SE. It is Op011 by setting it as 'AL'. It is delaying a result. Op011 The result of having been delayed is field DEC.DM.A.SE in a cycle 7. It is Op014 by setting it as 'SP'. It can teach a DM.A storage unit at the time of initiation.

He is trying for the specific register SP to maintain the engine-performance level of a true multiport machine, while the single collision of two values always arises in the predetermined cycle of arbitration, avoiding back track actuation. This will say that a functional unit can read only one value from a storage unit in each cycle, and means that one side of a collision result is always required in front of another side. Because of basis, it is Op011. A result is Op004. It is Op004 when it turns out that it is what is demanded more compared with a result. The number of two cycles, i.e., arbitration, can carry out cycle delay of the result until it will become more urgent than a collision result. In all cases, the need for the continuous re-scheduling of other actuation is avoided.

The number of the functional units which should be included, and selection of a type can be used sufficiently freely. Each unit of the specific gestalt of drawing 1 can combine all of the group of these functions with a functional unit more common multiple-purpose type to the maximum flexibility in scheduling, while being specified to some extent as performing the group of a specific function or a function. Since sufficient pipeline is permitted, I hear that it is desirable to have a certain amount of latency to all actuation of a predetermined unit as for the inconvenience of this type of unit, and it has it. This means that it is necessary to slow down actuation quicker than (for it to be like fixed generating) to the minimum working speed. In order to perform for example, a floating point multiplication function, a barrel shift function, or an I/O function, it is necessary to enable it to use the functional unit type specified more to it also for a predetermined application. since the unit same type as being used by the usual microprocessor is contained in the processor of this invention -- this contractor -- good -- arrangement -- that is, it is made to suit easily

Memory interface circuitry which has exception situation recovery Drawing 6 shows the configuration of a suitable memory interface circuitry to use it as a data memory unit DM of the processor of drawing 1 -4. Signal DM.G (guard bit), DM.A (address), and DM.I (write-in data) are received from the imitation multipoint memory 30 (refer to drawing 1). A storage management unit (MMU) supplies signal DM.O to a store circuit 30 through a circuit 100 while receiving these signals through a certain (not shown to drawing 1) exception situation recovery circuit 100. op code signal DM.OP (read-out/writing) is

received from an instruction register 46 (refer to drawing 1). MMU Large-scale off chip main memory MEM The interface to receive is offered and virtual memory address assignment well known by this contractor is performed especially in itself.

Signal DM.A and DM.I are not affected in the exception situation recovery circuit 100, but guard-bit DM.G is MMU as signal DM.G' simultaneously. Before making it pass, it is AND in the protection enforcement signal GF. It calculates. When Signal GF is in the condition of logic '1' (truth), signal DM.G' is equal to signal DM.G. However, '0' (false) is forced DM.G' when Signal GF is logic '0' (false). The protection enforcement signal GF is guard-bit AL.G of other functional units of the processor of drawing 1 , and RG.G. And BR.G (it does not explain), It is AND with the same approach. It calculates.

The exception situation recovery circuit 100 is a multiplexer 102, and the record/regenerative circuit RRC. It contains. Circuit RRC Playback mode signal REP A multiplexer 102 is controlled. REP -- between logic '0' and a multiplexer -- MMU from -- generated signal DM.O' is transmitted to the imitation multiport store circuit 30 as signal DM.O. however, REP Between logic '1', and record/regenerative circuit RRC from -- generated playback data signal DREP is transmitted to a store circuit 30 instead of DM.O'. therefore, playback mode signal REP the recording mode which is DM.O=DM.O' -- DM.O=DREP it is -- it distinguishes from a playback mode. Record / regenerative circuit RRC Signal DREC=DM[from the output section of a multiplexer 102] .O It takes out. (It characterized by signal GF= '1' and REP= '0') In normal operation, substantially, the broken-line box 100 becomes transparent and becomes result DM.G'=DM.G and DM.O=DM.O'. MMU It is each cycle and is main memory MEM. It is condition DM.G=, while reading value DM.O from location DM.A or being dependent on op code DM.OP. It rules over truly and is main memory MEM. The demand which writes value DM.I in location DM.A is received. MMU Record / regenerative circuit RRC Memory MEM from -- the event signal ETV (ETV=1) showing the availability of value DM.O by which reading appearance was carried out -- record / regenerative circuit RRC It supplies.

MMU TLB explained later again Supplying the loss signal TLBM to the input section of the OR gate 104, the output section of the OR gate is the exception situation detecting signal EXDET. Sequencer SEQ And it conveys to the instruction generating pipeline IIP (refer to drawing 1). The input section of another side of the OR gate 104 is used for receiving an exceptional event start signal from other functional units.

Signal EXDET The reset input section (R) of the set-reset flip-flop 106 which synchronizes once for every machine cycle is supplied. Instruction generating pipeline IIP Exception situation handling signal EXH And checkpoint signal CPT It is made to generate and they are record / regenerative circuit RRC about these signals. It supplies. Exception situation handling signal EXH Supplying the set input section (S) of a flip-flop 106, the output signal (Q) of a flip-flop 106 constitutes the protection enforcement signal GF simultaneously.

Drawing 7 is record / regenerative circuit RRC. It is shown more in a detail. Time amount counter TIC which synchronizes once for every machine cycle Checkpoint signal CPT It receives and the time amount signal TREC is generated. The time amount signal TREC is connected with Signal DREC, and it is the event memory ECM. The data input section (DIN) is supplied. Signal DM.O is the event memory EVM, if it assumes that it is width of face of 8 bits to the time amount signal TREC, since it has 32 bits. A data input has width of face of 40 bits. The time amount signal TREC is the checkpoint signal CPT and the event signal EVT again. And exception situation handling signal EXH It is the event count logical circuit EVC like. It is supplied. Logical circuit EVC It is the event memory EVM by each of address signal ADDR and the write-in enable signal WE. It controls.

Event memory EVM It has 40-bit data output DOUT, and is a logical circuit EVC about the 8 bits event time amount signal TREP of this output, and a 32-bit playback data signal. And latch 108 is supplied, respectively. latch 108 -- logical circuit EVC from -- the following event signal NXT supplied it synchronizes -- having -- simultaneous -- the latch output section -- playback data signal EREP is supplied to a multiplexer 102.

The configuration of the storage management unit to sufficiently known virtual-memory actuation and

detailed explanation of operation are not required for this contractor here because of an understanding of this invention. If it summarizes, it will be MMU of this gestalt. A translation lookaside buffer (TB or TLB) is included, and this translation lookaside buffer is Memory MEM about the virtual address (DM.A). Two or more entries mapped in a physical address are included. It is TLB only about the subset of the address which may exist. It can map in predetermined time amount and is TLB. When address DM.A can be mapped by the existing entry, the demand of read-out to address DM.A or writing can be fitted only in the inside of data memory unit latency (two cycle of this gestalt). When address DM.A cannot map by the existing entry, this needs to constitute the exception situation and needs to interrupt program execution. This so-called TLB About a loss, it is new (the existing entry is deleted) TLB. Since it is solvable by creating an entry, it is TLB. A loss is called important ["un-important"] or a "possible [recovery]" exception situation. MMU Signal TLBM=1 is used and it is TLB. While expressing generating of a loss, exception situation handling and a recovery process are started.

Example of activation which has the exception situation The schedule of drawing 8 is already carried out to two or more very long instruction word (refer to IW of drawing 2), and it shows in linearity the program fragmentation of the 2nd example memorized by program counter location PC=0 and refer to 1 or 2 or less. All actuation by which the schedule was carried out is not illustrated and it is main memory MEM in the case of activation. Five actuation which carries out 1 increment to the value memorized by the position is shown. Fragmentation does not use intentionally the value which already exists in the imitation multiport store circuit 30. Therefore, the machine condition of the relation in PC=0 exists in the register of the data memory unit DM (main memory MEM) and the register unit RG thoroughly. Furthermore, the schedule of the actuation which is not completed before PC=0 is not carried out in front of PC=0. Therefore, the checkpoint instruction which expressed to PC=0 by the asterisk '*' in drawing 8 can be shown. When flows of control "branch" in a checkpoint, the condition of this latter always is not required and this means that a medium history (the pipeline's content) is always the same at the time of checkpoint attainment.

If an entry is carried out to the fragmentation of PC=0 in the 2nd example, the pointer to the storage location of relation will be memorized by the register 0 of the register unit RG. Of operation 'rdreg' started by PC=0 copies the pointer from here to a temporary location t1 of the imitation multiport store circuit 30. In this case, t1 is received by PC(RG latency is 1) =1. Of operation 'read t1' started by PC=1 reads the value memorized in the storage location of relation in the 2nd temporary location t2. In this case, t2 is received by PC(DM latency is 2) =3. In the meantime, of operation 'constant(1)' started by PC=2 loads a delta value 1 to the 3rd (CO latency is 0) temporary location t3. Of operation 'aplus t2 t3' started by PC=3 permutes the sum of the value from memory t2 and an increment t3 by the 4th temporary location t4. In this case, t4 (AL latency is 1),

It receives by PC=4. Finally, of operation 'write t1 t4' started by PC=4 returns to the primary-storage location also showing a temporary location t1, and writes in the value t4 which carried out the increment. In this case, t4 reaches PC(TLB assuming that there is no loss) =6.

In case wave drawing 9 performs program fragmentation of the 2nd example of drawing 8 by PC=5, it shows the exception situation recovery process using drawing 6 in the case of assuming that the exception situation (for example, TBL loss) is detected, and the circuit of 7. Top line 'PC=' of drawing 9 expresses the program enumerated data of a configuration of generating from an IIP pipeline in the continuous cycle of a processor started by PC=B which has branching to checkpoint PC=0. The following, Signal CPT, TREC (time amount signal enumerated data are shown), EVT, EXDET, and GF and EXH And REP A wave is expressed. It sets at the time of wave-like initiation, and they are Signals CPT and EVT, EXDET, and EXH. And REP All are set to '0' and GF is set to '1'. These express normal operation (recording mode which does not deal with the exception situation). since the number signal TREC of hour meters is a former (not shown) checkpoint -- counting -- it becomes a cycle.

A checkpoint instruction is expressed to program memory by setting bit CP='1' as the instruction word (referring to drawing 2) of relation. With this gestalt, it is set as instruction PC=B which starts the branching actuation to a checkpoint rather than attached to checkpoint instruction itself in Bit CP. This means gives advantageous warning of each checkpoint of 5 cycle, and relaxation timing is checkpoint

signal CPT=1 by branching delay. The circuit which is generated or answers a checkpoint by the option is restrained. Naturally, it is signal CPT=1 when it is FALSE (BR.G=FALSE and/or GF=0) by which branching actuation is guarded. It does not generate. It is the checkpoint signal CPT to the last of cycle PC=L of the last of branching delay. At the time of initiation of the cycle which attached 'C', it sets up by the instruction generating pipeline (in logic "1"). In this case, it is referred to as PC=0 at the time of the beginning.

When a return needs to be carried out so that the exception situation may be followed, it is checkpoint signal CPT=1. Sequencer SEQ PC value of a checkpoint is made to memorize. checkpoint signal CPT=1 [moreover,] Time amount counter TIC in record / regenerative circuit RRC (drawing 7) zero -- resetting (TREC=0) -- an event -- counting -- logical circuit EVC Address ADDR -- zero -- setting up -- event memory EVM It clears effectively. The increment of the number signal TREC of hour meters is carried out in the following cycle, respectively, and it acts as a rule of thumb of the time amount progress from a checkpoint. Event signal EVT Value by which reading appearance was carried out from memory (being the gestalt of signal DM.0=DM.O')

It is MMU, when it can use first and the cycle (PC=3) which attached T is received. It sets up. event signal ETV=1 answering -- an event -- counting -- logical circuit EVC Event memory EVM the write-in enable signal WE -- setting up -- record data signal DREC=DM.O=DM.O' -- the "time amount stamp" of the gestalt of the number signal TREC of hour meters -- current -- event memory EVM equal to 3 The first location is made to memorize. According to this, the increment of the event memory address ADDR is carried out for preparation of storage of the following event.

It sets to E (PC=5) and is exception situation detecting-signal EXDET=1. It is TLB as shown. A loss or other exception situations occur. Thereby, a flip-flop 106 is made to generate guard enforcement signal GF=0, and activation of meantime PC=6 and PC=7 is continued. By these two cycles, actuation of the arbitration containing for example, of operation 'write t1' is started before an exception situation, and those activation is completed. Therefore, these and other actuation which are started before the exception situation is detected change a machine condition from the condition which exists in a checkpoint of having been specified good. Even if it starts new actuation in these two cycles, the actuation started by protection enforcement signal GF=0 after the exception situation arises does not affect a machine condition, consequently the machine condition is exact at time amount R.

It is used by a sequencer and the instruction generating pipeline, and the two-cycle delay between E and R is TLB. In the case of the exception situation of a loss, it is MMU. TLB Transfer control to the exception situation handling routine (PC=X) which sets up a new entry is performed.

This delay is shorter than the usual branching delay, and makes min disadvantageous profit of time amount to processing of the exception situation. Signal EXH=1 which answers carrying out a return in the guard enforcement signal GF '1', and generates the persistence time of an exception situation manipulation routine in time amount R It is marked and is the time amount counter TIC. Counting is stopped (TREC=7).

An exception situation manipulation routine is operated using the subsegment of addressable memory eternally and directly. By this means and suitable programming, exception situation manipulation-routine itself guarantees not generating absolutely the exception situation which is not important. reference Colwell which explained this means in itself, for example, already, etc. -- from -- it is known at this contractor. The actuation as which an exception situation manipulation routine is required is the approach of sufficiently known [contractor / this], and is MMU. It depends and other publications of these are not required for an understanding of the exception situation recovery indicated here. Exception situation handling can also be performed by specific hardware, without preparing clearly by the programmer. For example, 680XO(s) It sets to a processor and is TLB about specific "table Wolk" hardware. It prepares in order to process the exception situation of loss.

It goes to termination of an exception situation manipulation routine, and is Sequencer SEQ. pC value of the latest checkpoint (this example PC= 0) is recovered, and he is the instruction generating pipeline IIP about it. The instruction word IW which inputted, consequently was memorized by PC=0 is loaded to the time amount which attached C' by drawing 9 at an instruction register 46, and this starts rerun of the

program fragmentation of the 2nd example started from checkpoint PC=0. About the checkpoint signal CPT, they are record / regenerative circuit RRC. Time amount counter TIC It answers resetting the time amount signal TREC to zero, and sets up at the time of this cycle, and counting of the cycle from a checkpoint is also started with this checkpoint signal. an event -- counting -- logical circuit ECV the value DREC which reset the address ADDR of event memory to zero, and was recorded by T as a result, and time amount stamp TREP=3 -- event memory EVM It can use in the data output section DOUT. rerun -- hitting -- an event -- counting -- logical circuit EVC Playback selection-signal REP=1 It sets up. therefore, the multiplexer 102 -- MMU from -- the data path of generated signal DM.O' -- being interrupted -- latch 108 -- minding -- event memory EVM from -- generated playback data signal DREP is constituted. It sets to time amount T' of rerun, and is Circuit EVC. It detects that between the time amount stamps TREP remembered to be the present time amount signals TREC is equal, and is Signal NXT. It uses and the value recorded on the time amount E under original activation is inputted into latch 108. Therefore, value DM.O supplied to the imitation multiport store circuit 30 by time amount T' is the memory MEM of the original time amount T. It becomes a value and does not become the delta value memorized by of operation 'write t1 t4' under original activation of program fragmentation here. therefore, the value for which it depends on a former exact machine condition instead of the value depending on the actual machine condition bad specified by activation among the playback mode -- event memory EVM from -- it turns out that it is supplied. Setting to PC=6 under rerun, of operation 'write t1 t4' is only Memory MEM during original activation. Overwrite of the same value is only carried out to the memorized delta value. Since only an exact input is received from the machine condition under rerun (as), the exact output to a machine condition is guaranteed. It is suitable that the reason for explaining with the header according to the following individuals performs read-out and no circuit actuation into a playback mode actually.

It is Circuit EVC if it returns to drawing 8 and the example of activation of 9. It is a regenerative signal REP just before time amount R' (PC=8) rerun exceeding the point that activation was interrupted in order to process the exception situation. It resets. It is the event memory EVM about the event (data transfer) of everything but arbitration by this. Exact recovery is attained when it is one or more of other events of arbitration in which it can add to the event (up to Event T) recorded before, therefore rerun can be required from the same checkpoint.

Main memory MEM [in / in the case of a larger-scale processor, two or more data storage units DM are formed, and / each cycle] The receiving multi-access is made possible. the gestalt of such [naturally] operation -- record/regenerative circuit -- main memory MEM from -- each data path is required. Since the whole processor synchronizes and operates, it is Circuit RRC. The existing component TIC, for example, a time amount counter, It is sharable between two or the circuit beyond it. A storage unit besides these is the same main memory tooth space MEM from the memory according to other individuals, in order to increase the range of parallel processing. Additional parts are offered ideally. Since the register unit RG also includes a programmer visible machine condition, the description of exception situation recovery is also established. However, Hwu of the reference already quoted using the backup copy register since the amount of the condition of Unit RG was dramatically small With the usual approach which it reaches and was indicated for the example by patt, it is storage **** about the condition in each checkpoint.

Therefore, since it is brief, the exception situation recovery description of a register unit is not indicated further. when the number of registers cannot hold economically [and] to perfect backup copy as an alternative, this contractor understands that a register (even -- or **) unit output-data path (RG.O) can include record/regenerative circuit. [very]

Furthermore, in the gestalt of another operation, content itself of an imitation multiport memory can be included in the definition of a machine condition, and each memory unit (26 drawing 1) establishes the exception situation recovery description with the gestalt of a backup storage unit, or a record/regenerative circuit in this case according to the size of the storage unit concerned.

Record/regenerative circuit without a time amount stamping Drawing 10 shows other examples of record/regenerative circuit of drawing 7 , and is the time amount counter TIC in this case. It does not

prepare but is the event memory EVM about a time amount stamp value (TREC). It is not made to memorize using a data entry. This is the event (generated from MMU- drawing 6) signal EVT to time amount T' under playback which was carried out to the time amount T under activation of the beginning of a checkpoint instruction although the control circuit of the circumference of it was clearly simplified while decreasing the size of event memory. It is dependent on generating, therefore, other examples -- setting -- record/regenerative circuit -- event signal EVT in a recording mode "point which reproduces the value newly remembered that the event signal in a playback mode is expressed while remembering that it is expressed by the always new value DREC -- coming -- it can enter --" "point -- coming -- taking out -- " -- that is, a FIFO memory is offered effectively.

Naturally, it is the event storage control circuit EVC. While maintaining a number of a value of counting recorded on accuracy from the checkpoint and generating Address ADDR, in case rerun passes the point which activation interrupted for the former attempt, the change to a recording mode from a playback mode is performed.

Naturally, a system designer is the starting value and the event signal EVT of Address ADDR to the easy example of drawing 10 . It is necessary to satisfy not reproducing the value corresponding to [number / of manifestation] accuracy in both a recording mode and a playback mode which has already recorded the playback value on accuracy.

Constraint of the memory access of a playback mode Deformation of the gestalt of the above-mentioned implementation can be required of a actual system. For example, as already explained, instruction't2 <- read t1' is rerun in rerun of the program fragmentation (drawing 8 and 9) of the 2nd example, instruction'write t1 t4' is permitted after that, and the value written in with the same instruction during original activation is processed, and overwrite is carried out. While the inaccurate value by which reading appearance was carried out from the address t1 is disregarded, and while the value by which overwrite was carried out to the address t1 is exact, it is desirable to forbid the exact read-out actuation and the write-in actuation in a playback mode generally.

As for access actuation of the storage location, this reason has often carried out the trigger of other events, therefore a repeat [**** / un-] is performed. Actually, an I/O device like UART for serial communication is often addressed as the mere storage location ("memory by which the map was carried out"). However, in such a case, a value is transmitted twice to a remote device by actuation which writes the same value in the storage location twice. Instead of receiving character string'DATA', a remote device receives 'DAATA'. Similarly, signal transmission of the WORD is usually carried out at UART permitted by the processor, and the following alphabetic character which is not the same produces read-out actuation of the alphabetic character received from the UART register by the repeat of read-out. Therefore, when a remote device transmits train'DATA', a processor receives 'DAA'.

I/O which carried out the memory map Even when equipment does not exist, repeat memory read-out actuation can carry out the trigger of the event which a virtual storage system does not expect. For example, in activation of instruction't2 <- read t1' of the program fragmentation of the 2nd example, while accessing the storage location t1 at accuracy, data conversion is the event memory EVM. It is assumed that it was recorded. Furthermore, the page of the memory which includes a location t1 while processing the exception situation is TBL. It is assumed that it is thrown away in order to pack "new refer to the page." If rerun begins from a checkpoint, the value demanded is the event memory EVM. It is TLB, if storage read-out actuation in which it succeeded around the last time amount is performed even if recorded. The exception situation of a loss arises. Clearly, it is very desirable to have not asked for this situation, and to arrange also to this reason, so that memory access may be controlled to the validity in a playback mode.

Deletion of a control pipeline By giving the information of the timing of each type of actuation more detailed than an easy latency value to a compiler, an architect understands that a "control pipeline" can be deleted or it can be made to decrease depending on the case. Many delay circuits are deleted from a processor with this technique, and, thereby, a physical circuit dimension is decreased. Instead, a compiler uses the detailed information, and carries out the schedule of the different bit field relevant to desired actuation according to an individual, consequently each bit, i.e., a bit field, is not loaded to an

instruction register 46, and it is not provided for a functional circuit to the machine cycle demanded actually.

For example, in a predetermined cycle ($PC=i$), it is assumed that he wants to start the actuation which should be performed with the functional unit which has the latency of a four cycle. At the actual example of the processor explained here, it is the guard selection field ($--.G.RA$) to a unit $PC=i+OD$. It can be made to be able to generate, the op code can be generated in $PC=i+1$, and one or the argument selection field beyond it ($--.RA$) can be generated in $PC=i+2$ or $PC=i+3$. Destination control field ($DEC.--.SE$ and $DEC.--.WA$) are not required to the last cycle of a latency period, but can be generated in cycle $PC=i+4$.

Although such a means is known theoretically, actual already becoming complicated to a known exception situation processor is known. That reason is that the clear answer which asks and receives "whether the activation front stirrup completed this actuation after activation" does not exist any longer. this complexity -- 1990 year 1 [for example,] 'The CYDRA-5 Departmental Supercomputer' of COMPUTER of the moon -- Rau etc. -- reference is made. Refer to 'Clearly, handling an exception--' which follow 26 pages from 25-page item initiation especially. Therefore, an architect prevents from deleting control pipeline arrangement, therefore takes care not to decrease a circuit dimension by these considerations.

To it, according to the guard device of the equipment explained here, the clear answer to this question can be permitted, therefore an architect can delete control pipeline arrangement. If the exception situation occurs, when generating the guard selection field to pending actuation before the guard enforcement signal GF becomes active, while the exception situation is detected, it is necessary to consider each pending actuation completed only in that case. In this case, when generating other fields, it is important for the architect of the equipment which determines to remove control pipeline arrangement freely, and its scheduling compiler.

Since the timing of generating of the guard selection field is different about detailed activation of actuation of each different type, the architect of a processor needs to take care so that the time amount arrangement of the change between a playback mode and a recording mode may be appropriately carried out to each functional unit.

This contractor understands the approach of using the above-mentioned record/playback device circuit, its above, and other deformation to exception situation recovery of wide range data processing system other than the VLIW processor explained here.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

1. In Operating Data Processing System Controlled by Sequence of Instruction Which Forms Program (a) Said program is performed while specifying the number of the checkpoints of said program. A step including the data transfer of the 1st type depending on the machine condition that this activation was specified with the data transfer in said system, (b) The step which answers the exception situation produced during activation of this step (a), and interrupts said program execution, (c) In the data-processing-system actuation approach equipped with the step which reruns said program from the latest checkpoint after solution of said interruption Said step (a) includes record of said 1st type of each data transfer from a checkpoint recently [said]. Said step (c) The data-processing-system actuation approach characterized by including the playback of the recorded data which synchronized with said rerun instead of actual data transfer.
2. Said step (c) is the data-processing-system actuation approach of the claim 1 publication characterized by including the restart of usual activation of said program while recording said 1st type of data transfer according to said step (a), when said rerun reaches the point that said activation was interrupted for said step (b).
3. Claim 1 characterized by enabling it to complete those activation with instruction which transmitted control of said system to exception situation manipulation routine from said program, was delayed in transfer of said control as a result of interruption of activation of said step (b), and was generated before said exception situation, or data-processing-system actuation approach of two publications.
4. Data-processing-system actuation approach of claim 3 publication characterized by forbidding actuation started during said delay before transfer of control to said exception situation manipulation routine so that it may not have effect of arbitration in said specified machine condition.
5. For said 1st type of data transfer, said specified machine condition is the data-processing-system actuation approach of the claim 1 publication characterized by including transmitting read-out of a value and its value from the location of said data memory to the processing component of said system including the content of the data memory of said system.
6. Claim 1 which makes said specified machine condition subset of all machine conditions of said processing system, and is characterized by specifying said checkpoint at point of said program in case only said specified machine condition relates to next instruction of said program, or data-processing-system actuation approach given in two.
7. It is Data Processing System. A Means to Generate Sequence of Instruction Which Forms Program on the occasion of Activation by this System, A means to identify the checkpoint in said program While performing said program, data transfer of the 1st type is performed. The 1st data path for which each of such data transfer depends on the machine condition that said system was specified, A means to answer the exception situation, and to interrupt said program execution In data processing system equipped with a means to rerun a program from the recently checkpoint after solution of said exception situation Said system A means to connect with said data path and to record each of said 1st type of data transfer from a checkpoint recently [said] While interrupting said account data path of rerun Nakamae Data processing

system characterized by having further a means to reproduce the recorded data transfer instead of actual data transfer synchronizing with said rerun.

8. Data processing system of claim 7 publication characterized by including means to reboot said playback means if said rerun reaches point that said activation was interrupted before while restoring said data path.

9. Claim 7 characterized by including means to make it make those activation complete by actuation which said exception situation was answered, control of said system was transmitted to exception situation manipulation routine from said program, and transfer of this control was delayed, and was started before said activation, or data processing system given in eight.

10. Data processing system of the claim 9 publication characterized by forbidding the actuation started during the delay before a transfer of the control to said exception situation manipulation routine, and including a means to make it such actuation not affect said specified machine condition.

11. For said data path, said specified machine condition is the data processing system of the claim 7 publication characterized by connecting the read-out port of said data memory to the processing component of said system including the content of the data memory of said system.

12. Said data memory is the data processing system of the claim 11 publication characterized by having a virtual storage system.

13. The claim 7 characterized by including two or more functional units which are controlled by each field of each instruction and perform each actuation to juxtaposition, or data processing system given in eight.

14. it be the data processing system of the claim 13 publication which carry out [also include a means forbid the actuation which be complete before detection of said activation , and] as the description by prepare each guard signal which prevent those actuation that affect said specified machine condition in spite of generating of a suitable instruction facultative in some of said functional units [at least] , and said system answer detection of said activation , and force each guard signal of said functional unit in non-conditions .

15. In Performing Predetermined Actuation, Control Said Processing System by Field of One or More Instructions. Decrease control pipeline arrangement and at least one guard signal which prevents the predetermined actuation which affects said system at said specified machine condition facultative is prepared. Said field includes the guard selection field which chooses one guard signal from the guard signal which can use plurality to predetermined actuation. By referring to whether said guard selection field was generated before said interruption in contrast with the field of other arbitration The claim 7 characterized by specifying actuation so that it may generate before the interruption produced according to the exception situation, or data processing system given in eight.

16. Data processing system of the claim 14 publication characterized by controlling the read-out actuation and the write-in actuation to said data memory during rerun of an instruction.

In Said Record Means List, 17. Said Interruption and Playback Means The multiplexer connected to said data path by having the output section linked to the 1st input section linked to said storage means, and said processing means, The event memory circuit which has the data input section linked to said data path, the data output section linked to the 2nd input section of said multiplexer, and the address input section, While connecting said storage means to said processing means by said multiplexer among a (i) recording mode While making said event memory memorize each value transmitted to said processing means from said storage means and connecting the data output section of said event memory to it by said multiplexer among the (ii) playback mode at said processing means The claim 7 characterized by having the control circuit which reproduces the value memorized by said recording mode before by said event memory, or data processing system given in eight.

[Translation done.]

(2)

特表平9-509515

【特許請求の範囲】

1. プログラムを形成する命令のシーケンスによって制御されるデータ処理システムを作動させるに当たり、

(a) 前記プログラムのチェックポイントの数を規定するとともに前記プログラムを実行し、この実行が前記システム内のデータ転送を伴い、規定されたマシン状態に依存する第1タイプのデータ転送を含むステップと、

(b) このステップ(a)の実行中生じる例外事態に応答して前記プログラムの実行を中断するステップと、

(c) 前記中断の解決後最近のチェックポイントから前記プログラムを再実行するステップとを具えるデータ処理システム作動方法において、

前記ステップ(a)は、前記最近チェックポイントから前記第1タイプの各データ転送の記録を含み、前記ステップ(c)は、実際のデータ転送の代わりに、前記再実行に同期した、記録されたデータの再生を含むことを特徴とするデータ処理システム作動方法。

2. 前記ステップ(c)は、前記実行が前記ステップ(b)で中断された点に前記再実行が到達すると、前記ステップ(a)に応じて前記第1タイプのデータ転送を記録する間の前記プログラムの通常の実行の再開を含むことを特徴とする請求の範囲1記載のデータ処理システム作動方法。

3. 前記ステップ(b)の実行の中断の結果、前記システムの制御を、前記プログラムから例外事態処理ルーチンに転送し、前記制御の転送を遅延して、前記例外事態前に発生した命令により、それらの実行を完了できるようにすることを特徴とする請求の範囲1又は2記載のデータ処理システム作動方法。

4. 前記例外事態処理ルーチンへの制御の転送前の前記遅延中に開始した動作を、前記規定したマシン状態に任意の影響を有しないように禁止することを特徴とする請求の範囲3記載のデータ処理システム作動方法。

5. 前記規定されたマシン状態は、前記システムのデータメモリの内容を含み、前記第1タイプのデータ転送は、前記データメモリの位置からの値の読出し及びその値を前記システムの処理素子に転送することを含むことを特徴とする請求の範囲1記載のデータ処理システム作動方法。

(3)

特表平9-509515

6. 前記規定されたマシン状態を、前記処理システムの全マシン状態のサブセットとし、前記チェックポイントを、前記規定されたマシン状態のみが前記プログラムの次の命令に関連する場合の前記プログラムの点に規定することを特徴とする請求の範囲1又は2記載のデータ処理システム作動方法。

7. データ処理システムであって、

このシステムによる実行に際しプログラムを形成する命令のシーケンスを発生させる手段と、

前記プログラム中のチェックポイントを識別する手段と、

前記プログラムを実行する間第1タイプのデータ転送を行い、このようなデータ転送の各々が、前記システムの規定されたマシン状態に依存する第1データ経路と、

例外事態にตอบสนองして前記プログラムの実行を中断する手段と、

前記例外事態の解決後の最近チェックポイントからプログラムを再実行する手段とを具えるデータ処理システムにおいて、

前記システムは、

前記データ経路に接続して、前記最近チェックポイントから前記第1タイプのデータ転送の各々を記録する手段と、

前記再実行中前記データ経路を中断するとともに、実際のデータ転送の代わりに、前記再実行に同期して、記録されたデータ転送を再生する手段とを更に具えることを特徴とするデータ処理システム。

8. 前記データ経路を復元するとともに、前記実行が以前に中断された点に前記再実行が到達すると前記再生手段を再起動する手段も含むことを特徴とする請求の範囲7記載のデータ処理システム。

9. 前記例外事態にตอบสนองして、前記システムの制御を前記プログラムから例外事態処理ルーチンに転送し、この制御の転送を遅延させて、前記実行前に開始した動作によりそれらの実行を完了させるようにする手段も含むことを特徴とする請求の範囲7又は8記載のデータ処理システム。

10. 前記例外事態処理ルーチンへの制御の転送前の遅延中に開始された動作を禁止して、このような動作が前記規定されたマシン状態に影響を及ぼさないよう

(4)

特表平9-509515

にする手段も含むことを特徴とする請求の範囲9記載のデータ処理システム。

11. 前記規定されたマシン状態は、前記システムのデータメモリの内容を含み、前記データ経路は、前記データメモリの読出しポートを前記システムの処理素子に接続することを特徴とする請求の範囲7記載のデータ処理システム。

12. 前記データメモリは仮想記憶システムを具えることを特徴とする請求の範囲11記載のデータ処理システム。

13. 各命令の各々のフィールドによって制御されて各々の動作を並列に実行する複数の機能ユニットを含むことを特徴とする請求の範囲7又は8記載のデータ処理システム。

14. 前記機能ユニットの少なくとも一部に、適切な命令の発生にもかかわらず前記規定されたマシン状態に影響を及ぼすそれらの動作を条件的に防止する各々のガード信号を設け、前記システムは、前記実行の検出に応答して前記機能ユニットの各々のガード信号を未条件的に強制することにより前記実行の検出前に完了しない動作を禁止する手段も含むことを特徴とする請求の範囲13記載のデータ処理システム。

15. 所定の動作を実行するに当たり、前記処理システムを一つ以上の命令のフィールドによって制御して、制御パイプライン配置を減少させ、前記システムに、前記規定されたマシン状態に影響を及ぼす所定の動作を条件的に防止する少なくとも一つのガード信号を設け、前記フィールドは、所定の動作に対して複数の利用できるガード信号から一つのガード信号を選択するガード選択フィールドを含み、前記ガード選択フィールドを、他の任意のフィールドとは対照的に、前記中断前に発生させたか否かを参照することにより、例外事態によって生じた中断前に発生するように、動作を規定するようにしたことを特徴とする請求の範囲7又は8記載のデータ処理システム。

16. 命令の再実行中、前記データメモリに対する読出し動作及び書込み動作を抑制するようにしたことを特徴とする請求の範囲14記載のデータ処理システム。

17. 前記記録手段並びに前記中断及び再生手段は、

前記記憶手段に接続した第1入力部及び前記処理手段に接続した出力部を有す

(5)

特表平9-509515

ることにより前記データ経路に接続したマルチプレクサと、

前記データ経路に接続したデータ入力部、前記マルチプレクサの第2入力部に接続したデータ出力部及びアドレス入力部を有する事象メモリ回路と、

(i)記録モード中、前記マルチプレクサにより前記記憶手段を前記処理手段に接続するとともに、前記事象メモリに、前記記憶手段から前記処理手段に転送した各値を記憶させ、(ii)再生モード中、前記マルチプレクサにより前記事象メモリのデータ出力部を前記処理手段に接続するとともに、前記事象メモリにより、前記記録モードで以前に記憶された値を再生する制御回路とを具備することを特徴とする請求の範囲7又は8記載のデータ処理システム。

(6)

特表平9-509515

【発明の詳細な説明】

データ処理システムの例外事態回復

発明の分野

本発明は、データ処理システムの例外事態回復方法及び装置に関するものである。本発明は特に、プログラムを形成する命令のシーケンスによって制御されるデータ処理システムを作動させるに当たり、

(a) 前記プログラムのチェックポイントの数を規定するとともに前記プログラムを実行し、前記変換が前記システム内のデータ転送を伴い、規定されたマシン状態に依存する第1タイプのデータ転送を含むステップと、

(b) このステップ(a)の実行中生じる例外事態に応答して前記プログラムの実行を中断するステップと、

(c) 前記中断の解決後最近のチェックポイントから前記プログラムを再実行するステップとを具えるデータ処理システム作動方法に関するものである。

また、本発明は、データ処理システムであって、

このシステムによる実行に際しプログラムを形成する命令のシーケンスを発生させる手段と、

前記プログラム中のチェックポイントを識別する手段と、

前記プログラムを実行する間第1タイプのデータ転送を行い、このようなデータ転送の各々が、前記システムの規定されたマシン状態に依存する第1データ経路と、

例外事態に応答して前記プログラムの実行を中断する手段と、

前記例外事態の解決後の最近チェックポイントからプログラムを再実行する手段とを具えるデータ処理システムにおいて、

前記システムは、

前記データ経路に接続して、前記最近チェックポイントから前記第1タイプのデータ転送の各々を記録する手段と、

前記再実行中前記データ経路を中断するとともに、実際のデータ転送の代わり

に、前記再実行に同期して、記録されたデータ転送を再生する手段とを更に具え

(7)

特表平9-509515

ることを特徴とするデータ処理システムに関するものである。

このような方法及びシステムは、1987年12月に刊行されたIEEE Transactions on Computers, Vol.C-36, No.12のW.-M.W.Hwu及びY.N.Pattによる“Checkpoint Repair for High-Performance Out-of-Order Execution Machines”の1469～1514頁から既知である。

関連技術の説明

データ処理システムは、プログラムを実行する間に生じる種々の「例外事態」に調和して対処する必要がある。例外事態は、例えば、中断、計算のオーバーフロー及びページフォールトを含む。(外部の、非同期事象により生じた)中断を、中断が処理された後セーブしかつ回復することができる良好に規定された状態にシステムが到達するまで、それら中断の操作を遅延させるとこにより処理することができる。これは、例えば68000 マイクロプロセッサファミリーで行われ、この場合、中断はマイクロ構造境界でのみ処理される。それに対して、計算のオーバーフローは、プログラムの一部の実行に対して重大なものであると見なすことができ、その結果、マシン状態とは関連がなくなる。

例外事態の他の範疇は、通常のプログラムの実行の間に発生し、この場合、制御は、ソフトウェアからハードウェアに付与される。有効な処理が要求される主な例は、仮想記憶システムの「ページフォールト」であり、この場合、アドレス指定可能な所定の部分のみが、所定の時に瞬時にアクセス可能となる。他の例は、実行されていないOPコードをエミュレートするトラップ、記憶管理ユニット(MMU)がテーブルウォークハードウェアを有しない場合のシステムのTLB ミストラップ、及びIEEE浮遊点の計算における特別の場合の処理を含む。このタイプの例外事態を、マシンの状態がどのようなものであっても瞬時に処理し、迅速かつ信頼性を有するように処理する必要がある、すなわち、仮想アドレス指定を有する大容量メモリのユーティリティを損失する。例えば、MC68020 では、プロセッサの関連の状態を、スタックしてセーブするとともに、例外事態を処理した後に回復させる必要がある。マシンは、その動作中決定性を有するので、有効であることがわかる状態にマシンを回復させることにより、あたかも例外事態が決して発

(8)

特表平9-509515

生しないように、あらゆる実用的な目的に対して、マシンが正確に動作を再開するようにする。

より高い性能のプロセッサは、より濃密なパイプライン配置になる傾向にある。追加のハードウェアデータ経路が要求されるので、このようなマシンの状態は非常に大きくなり、状態の大部分は、アクセスに対して高価なパイプライン段に存在する。したがって、実際にはプログラム中の特別な場合の「チェックポイント」を規定するようになる。実行する間にチェックポイントに到達すると、将来の動作に関連するものである限り、バックアップコピーがマシン状態を形成する。例外事態を処理した後、プロセッサの状態を、最近のチェックポイントのものに復元させ、実行をチェックポイントから再開する必要がある。

チェックポイントを用いる例外事態回復は、1987年12月に刊行されたIEEE Transactions on Computers, Vol.C-36, No.12のW.-M.W.Hwu及びY.N.Pattによる "Checkpoint Repair for High-Performance Out-of-Order Execution Machines" の1469~1514頁に記載されている。Hwu 及びPattは、小レジスタファイルのバックアップコピーを維持し、マシンをチェックポイントの状態に復元する方法及び装置を説明している。バックアップコピーは、多くの場合主メモリの全容量を含む、大容量のマシン状態に対して実用的でない。したがって、Hwu 及びpattは、最近チェックポイントから生じるメモリへのデータ転送により上書きされた値をセーブし、チェックポイントから実行を再開する前に状態を復元するためにこの値を書き込む方法を説明している。セーブした値は非常に小さいスペースを占有するが、これら値の再書き込みは、時間を浪費し、処理システムの全体に亘るパフォーマンスが低下するおそれがある。

チェックポイントリトライシステムは、欧州特許明細書第0348652 号にも記載されている。このシステムでは、キャッシュメモリに書き込まれた値が、チェックポイントを指定する命令が終了するまで特定のバッファに保持され、これにより、例外事態が生じていないことがわかるまで状態変化を遅延させる。

並列実行及び濃密なパイプライン配置に大幅に依存する処理装置のタイプは、「非常に長い命令ワード」(VLIW)タイプのプロセッサである。「理想VLIW」プロセッサ及びTRACE プロセッサとして既知の実用VLIWプロセッサは、1987年10月5

(9)

特表平9-509515

～8 日に、カリフォルニア週のパロアルト(Palo Alto)のArchitectual Support for Programming Languages and Operating Systems (ASPLOS II)のProceedings of the Second International Conferenceの180 ～192 頁のR.P.Colwell 等による'A VLIW Architecture for a Trace Scheduling Compiler' の題の論文に記載されている。例外事態回復に関して、Colwell 等は、一旦例外事態が解決されると、中断されたメモリ動作を再び行う「ヒストリキュー機構」をあいまいに説明している。

VLIWアーキテクチャを有するビデオ信号プロセッサは、上記関連する欧州特許明細書題325,310号(PHN 12.376)に記載されている。ビデオ信号処理の特定の用途では、物理メモリを用いた小サイクリックプログラムが使用され、その結果、復元可能な例外事態の問題が生じない。より一般的な処理装置の同様な構成は、1988年10月3 ～5 日のComputer Design のProc.IEEE International Conferenceの本発明者による'CREATE-LIFE: A Design System for High Performance VLSI circuits' 及び欧州特許明細書題479,390 号(PHA 1209)に記載されている。一般的な処理用途にこの構造を用いる場合、大容量の仮想メモリを要求するのが望ましいので、例外事態の問題が生じる。回路寸法及び/又はパフォーマンスの損失に関して、既知の回復システムを用いる不利益は、VLIWプロセッサで特に厳しくなることが確認されている。

欧州特許明細書479,390 号に記載された、VLIWプロセッサを説明する他の文献は、(場合に応じて)それに適合させる例外事態回復の方法を言及していない。

既知の例外事態回復システムに関連する他の問題は、これらシステムは「制御パイプラインの削除」を容易に行うことができず、特にVLIWプロセッサにおいて所定のパフォーマンスレベルに対してそれ以外の方法で回路寸法を減少させる方法が必要となる。

発明の要約

本発明の目的は、パフォーマンスの不利益及び/又は回路寸法を減少させる他の例外事態回復方法を提供することである。本発明の目的は、特に、VLIWプロセッサのような濃密なパイプライン処理システムを使用するのに適切な例外事態回

(10)

特表平9-509515

復方法を提供することである。

本発明は、プログラムを形成する命令のシーケンスによって制御されるデータ処理システムを作動させるに当たり、上記ステップ(a)、(b)及び(c)を具え、前記ステップ(a)は、前記最近チェックポイントから前記第1タイプの各データ転送の記録を含み、前記ステップ(c)は、実際のデータ転送の代わりに、前記再実行に同期した、記録されたデータの再生を含む方法を提供する。

チェックポイントにおける再実行前のマシン状態を復元する代わりに、本発明による方法は、現在のマシン状態を無視し、例外事態発生前の正確なマシン状態から得られる記録された値を構成する。Hwu及びpatt方の小回路寸法が維持され、同時に、正確なマシン状態の復元を含むパフォーマンスの不都合は除去される。既知の方法は、再実行に任意の影響を及ぼすか否かにかかわらず、チェックポイントから上書きされた全ての値を復元する。それに対して、本発明による方法は、本来の実行に正確に用いられた、したがって再実行に要求される値のみを記憶及び再生する。

処理装置のようなプログラミングに当たり、パイプライン段の内容を含む非常に限定された「プログラマ可視」状態を規定する。この際、チェックポイントは、この可視状態のみが将来の動作に関連するプログラムの点を規定する。これを、プログラムをチェックポイント指定したフラグメントに分割することによって配置することができ、このようにチェックポイント指定したフラグメントの各々は、一連の動作中一つのあり得るブレデッサのみを有する。換言すれば、制御の流れをチェックポイントで好適に分岐させる。分岐を回避できない特定の場合には、可視状態のみが(パイプラインが空である)分岐点に関連するようにプログラムをスケジューリングすることができ、すなわち正確な状態を満足する手段を設ける。

前記ステップ(c)は、前記実行が前記ステップ(b)で中断された点に前記再実行が到達すると、前記ステップ(a)に応じて前記第1タイプのデータ転送を記録する間の前記プログラムの通常の実行の再開を含んでもよい。この方法により、同一チェックポイントに続く他の任意の例外事態に自動的に対処すること

(11)

特表平9-509515

ができる。

例外事態を解決するのに要求されるステップは、必要な場合には特定のハード

ウェア及び／又はソフトウェア要素を含む当業者には既知の方法で、例外事態の性質に依存する。これらステップは、本発明の一部を形成しない。

これは、前記ステップ（b）の実行の中断の結果、前記システムの制御を、前記プログラムから例外事態処理ルーチンに転送し、前記制御の転送を遅延して、前記例外事態前に発生した命令により、それらの実行を完了できるようにする例で説明した。同時に、例外事態処理ルーチンに対する制御の転送前の遅延で開始した動作を禁止して、規定されたマシン状態にこれら動作が影響を及ぼさないようにすることができる。これら手段を、通常の例外事態が中断の点から再開する際にマシン状態が正値となるよう必要に応じて用いることができる。

特に、ハードウェアを用いることなく、LIFEタイプのVLIWプロセッサの「ガード」ファシリティを用いて、動作を禁止することができる。このファシリティ（又はこれと同等のもの）を用いることにより、従来の例外事態回復システムでこれが生じるという問題なく、パイプライン配置の制御を削除することもできる。

本発明は、上記本発明による方法に従う動作に適切なデータ処理システム及び装置を提供する。

図面の簡単な説明

本発明の実施の形態を、図面を参照して詳細に説明する。

図1は、本発明による処理装置の線形形態を示す。

図2は、図1の装置の非常に長い命令ワードのフォーマットを示す。

図3は、図1の装置のイミテーションマルチポートメモリの一部を示す。

図4は、第1例のプログラムフラグメントの動作を示す。

図5は、衝突の問題を含む図1の装置に対する第1例のプログラムフラグメントのスケジューリングを示す。

図6は、例外事態検出及び回復回路を含む処理装置の一部を示す。

図7は、処理装置の記録／再生回路をより詳細に示す。

図8は、第2例のプログラムフラグメントのスケジューリングを示す。

(12)

特表平9-509515

図9は、第2例のプログラムフラグメントの実行中の例外事態回復回路の動作を示す。

図10は、処理装置の記録/再生回路の他の例を詳細に示す。

処理装置の大要

図1の処理装置は、コンスタントユニットCO、計算及び論理ユニットAL、レジスタユニットRG、データメモリユニットDM並びに分岐ユニットBRを含む5個の機能ユニットに接続されたイミテーションマルチポート記憶回路30を具える。記憶回路30内に、記憶ユニット26と、マルチプレクサ27と、関連のマルチプレクサ29を有する特定レジスタSPとを線形的に図示する。このような記憶回路及びプロセッサの動作の原理は、欧州特許明細書第479390号(PHA1209)に記載されており、スラベンバーグ及びラブロスによって参照することができ、双方とも既に説明した。これらの原理は本発明の一部を形成せず、簡略のためにそれらの記載は省略した。

シーケンサSEQ及び命令発生パイプラインIIPを具えるグローバルコントローラ41を設ける。シーケンサSEQは、命令発生パイプラインIIPに供給されるプログラムカウンタワードPCを各サイクルで発生させる。命令発生パイプラインIIPをオフチップとすることができ、これは、実行プログラムを形成する非常に長い命令ワードが符号化されるプログラムメモリを含む。所定の数のサイクルの遅延後、命令発生パイプラインIIPは、オンチップ命令レジスタ46に、プログラムメモリのアドレスPCに記憶された非常に長い命令ワードIWを供給する。

レジスタ46の命令ワードIWは、宛先制御フィールドDECと、個別の機能ユニットCO,AL,RG,DM及びBRをそれぞれ制御する複数のフィールドCOC,ALC,RGC,DMC及びBRCと、チェックポイントをマークする別のビットCPとを具える。説明したプロセッサは非常に簡単な例であるが、命令ワードIWのビットの総数は163である。

図2は、命令ワードIWのフィールドをより詳細に示し、これらフィールドは、フィールドの下に付した各フィールドのビット数を有する。これらの種々のフィールドの目的は、プロセッサの構成及び動作の以下の詳細な説明から明らかにす

(13)

特表平9-509515

ることである。チェックポイントマーキングビットCPの目的は、図面の図6～10を参照して以下詳細に説明することである。

コンスタントユニットCOは、オペラント入力部を有しなく、記憶回路30の入力部に接続した32ビットの結果出力部(CO.O)を有する。コンスタントユニットCO

は、実際には、命令ワードIW(図2)の32ビットフィールドCOCとコンスタントユニットCOの出力部との間の直接接続を有する。これにより、コンスタントユニットCOは、記憶回路30を介して、命令ワードの範囲内から他のユニットに一定値を供給することができる。

計算及び論理ユニットALは、記憶回路30内の各メモリユニットに接続した1ビットのガード値入力部(AL.G)及び二つ(左右)の32ビットの変数入力部(AL.L及びAL.R)を有する。ユニットALは、記憶回路30に逆接続した32ビットの出力部(AL.O)を有し、命令ワードIWの16ビットフィールドALCによって制御される。

図3は、計算及び論理ユニットALに関連するイミテーションマルチポート記憶回路30の一部を詳細に示す。記憶回路30内には、個別の記憶ユニット(RAM)48, 50及び52を、オペラントAL.G, AL.L及びAL.Rにそれぞれ関連させる。更に他の通常の処理装置と対比すると、個別の言込みイネーブル信号が、図1～9の装置中の記憶ユニットに対して供給されない。その代わりに、少なくとも一つの存在しない、すなわち少なくとも一つの書き込み不可能な記憶位置を、その記憶ユニットに対する書き込みが所望されない際にアドレス指定することができる。言込みイネーブルビットの値がスケジューリングの時に既に既知であると仮定すると、結果が特定の記憶ユニットの位置に割り当てられる際に、命令ワードIWの幅を、これらイネーブルビットが存在しないことにより減少させる。

ガードビットAL.Gに対する記憶ユニット48は、説明したような一つの書き込み不可能な位置を含む、ブール値(TRUE/FALSEビット)を記憶するための8個の1ビット位置を有する。記憶ユニット48に対する3ビット読出しアドレスは、命令レジスタ46により、命令ワードIWの制御フィールドALCの3ビットフィールドAL.G.RA(図2)から供給され、記憶ユニットの1ビット読出しポート54は、ガードビットAL.Gを、アドレス指定された記憶位置からユニットALに供給す

(14)

特表平9-509515

る。

マルチプレクサ56は、種々の機能ユニットの結果出力部(CO.I,AL.O等)に接続された32ビットの最小重要ビット(1sb)に接続した複数の1ビットの入力部を有する。マルチプレクサの追加の入力部を、追加の32ビットバスライン58の1sbに接続し、これは、遅延された結果出力SP.Oを、特定レジスタSPの形態の遅延

素子から搬送する。このタイプのVLIWの衝突を解決する遅延素子を設けて使用すると、上記欧州特許明細書第479,390号(PHA1209)の課題を形成する。ガードビットに対して1sbを使用すると、例えばC言語のプログラミングを含む所定のプログラミング規定との互換性を有するようになる。他の状況では、最重要ビット(msb)又は他のビットを、より便利に選択することができる。

マルチプレクサ56は、記憶ユニット48の書込みポートに接続した1ビットの出力部60を有する。マルチプレクサ56は、命令ワードIWの宛先制御フィールドDECの3ビットの選択フィールドAL.G.SEによって制御される。記憶ユニット48の書込みポートに対する3ビットの書込みアドレスは、宛先制御フィールドDECの隣接する3ビットフィールドAL.G.WAによって付与される。

同様に、32ビットの記憶ユニット50(AL.Lに対する)及び(AL.Rに対する)52はそれぞれ、関連の32ビットのマルチプレクサ62及び64を有する。各々は、一つの書込み不可能なロケーションを含む16個の位置を有する。宛先制御フィールドDECは、マルチプレクサ62及び64をそれぞれ制御する二つの3ビット選択フィールドAL.L.SE及びAL.R.SEと、記憶ユニット50及び52の各々に対して4ビットの書込みポートアドレスを供給する二つの4ビットフィールドAL.L.WA及びAL.R.WAとを含む。制御フィールドALCの二つの4ビットフィールドAL.L.RA及びAL.R.RAは、記憶ユニット50(AL.Lに対する)及び(AL.Rに対する)52の読出しポートに対する読出しアドレスを発生させる。

計算及び論理ユニットALに対するopコードAL.OPは、ユニットALに対する制御フィールドALCの5ビットフィールドによって付与される。各サイクルにおいて、opコードAL.OPは、結果AL.Oを発生させるためにアーギュメントAL.L及びAL.Rに対してどの動作(加算、減算、比較、AND、OR等)実行すべきかを決定する。

(15)

特表平9-509515

図1に戻ると、レジスタユニットRGは、各々が32ビットの16個の頻繁に用いられるプログラム変数に対するオンチップ記憶装置を提供する。ユニットRGは、ガードビットRG.G及び32ビットのデータ入力RG.Iに対する記憶回路30内の関連の記憶ユニットを有する。これら記憶ユニットは、各々が一つの書込み不可能な位置を含む四つの位置のみを有する。データ出力(RG.O)を、各バスライン(RG.O、

図3参照)を形成する記憶回路30の入力部に接続する。宛先制御フィールドDECは、関連のマルチプレクサを制御する二つの3ビットの選択フィールドRG.G.SE及びRG.I.SEと、各記憶ユニットの四つの位置の間で選択する二つの2ビットの書込みアドレスフィールドRG.G.WA及びRG.I.WAとを含む。

9ビットのレジスタユニット制御フィールドRGB内で、二つの2ビットアドレスRG.G.RA及びRG.I.RAを、各記憶ユニットから読み出すべき値RG.G及びRG.Iの値の位置を選択するために設ける。5ビットのopコードフィールドRG.OPは、レジスタの読出し動作及びレジスタの書込み動作との間で選択する1ビットと、ユニットRG内の16個のレジスタ間で選択する4ビットとを有する。したがって、レジスタユニットRG内の変数の位置は、コンパイラによって固定される。

データ記憶ユニットDMは、少なくとも概念的には、変数に対して多量の記憶を含む。ユニットDMは、実際には、図6を参照して後に詳細に説明するように、読出し/書込みメモリのオンチップバンクに対するインタフェースを具える。イミテーションマルチポート記憶回路30内では、個別の記憶ユニットを、ガードビットDM.G、32ビットのアドレスワードDM.A及び32ビットの入カデータワードDM.Iに対して設ける。これら記憶ユニットは、各々が書込み不可能な位置を含む256個の位置を有する。データ記憶ユニットDMは、各バスライン(DM.O、図3も参照)を形成する記憶ユニット30の入力部に接続された32ビットのデータ出力(DM.O)を有する。宛先制御フィールドDECは、記憶回路30内のマルチプレクサを制御する三つの3ビット選択フィールドDM.G.SE、DM.A.SE及びDM.I.SEと、関連の記憶ユニットの各々の256個の位置間で選択する三つの8ビット書込みアドレスフィールドDM.G.WA、DM.A.WA及びDM.I.WAとを含む。

25ビットのデータ記憶ユニット制御フィールドDMC内に、三つの8ビット読出

(16)

特表平9-509515

しアドレスDM.G.RA,DM.A.RA 及びDM.I.RA を、それらの各記憶ユニットから読み出すべき値DM.G,DM.A 及びDM.Iの位置を選択するために設ける。1ビットのOPコードフィールドDM.OP は、記憶の読出し動作及び記憶の書込み動作との間で選択する1ビットを含む。アドレスワードDM.Aは、データワードDM.Iの書込み又はデータワードDM.Oの読出しに対するアドレスを規定する。これにより、プログラムの実行中、 2^{32} の位置をアドレス指定するのに十分な32ビットのアドレスの動的

計算を行うことができる。

分岐ユニットBRにより、プログラム実行中に分岐するように修正すべきシーケンサSEQ によって、発生した命令アドレスPCのシーケンスを許容することができる。ユニットBRは、記憶回路30内に、ガードビットBR.G及び32ビットの宛先アドレスBR.Aに対する関連の記憶ユニットを有する。これら記憶ユニットは、各々が書込み不可能な位置を含む四つの位置を有する。宛先制御フィールドDEC は、関連のマルチプレクサを制御する二つの3ビット選択フィールドBR.G.SE 及びBR.A.SE と、各記憶ユニットの四つの位置間で選択する二つの2ビット書込みアドレスフィールドBR.G.WA 及びBR.A.WA とを含む。

4ビット分岐ユニット制御フィールドBRC 内に、二つの2ビット読出しアドレスBR.G.RA 及びBR.A.RA を、各記憶ユニットから読み出すべき値BR.G及びBR.Aの位置を選択するために設ける。分岐ユニットBRは、ガードビットBR.G及び宛先アドレスBR.AをシーケンサSEQ に簡単に通過させる。OPコードフィールドBR.OR は必要でない。シーケンサSEQ 内で、ガードビットBR.Gのブール値がテストされる。BR.Gが真である場合、分岐を実行すべきであり、かつ、シーケンサSEQ は、記憶回路30から受信した宛先アドレスBR.Aに等しいPCの次の値を作成する。BR.Gが偽である場合、シーケンサSEQ は単にPCの値をPC+1に増分し、記憶されたプログラムの次の命令のアドレス指定を行う。

したがって、論理結果AL.OをBR.Gに対する記憶ユニットに指導する(1sb の真又は偽)ことにより、条件的な分岐を行うことができる。(常に行われる又は決して行われない)条件的でない分岐を、一定結果CO.Oを、BR.Gに対する記憶ユニットに指導することにより、すなわち、記憶ユニットが常に真又は常に偽となる

(17)

特表平9-509515

書込み不可能な位置の配置を行うことにより実行することができる。この後者の追加は、既に説明したように各記憶ユニットの少なくとも一つの位置が常に書込み不可能であるので、本発明の実施の形態において特に便利である。(分岐が決して行われない) ガードビットBR.G=FALSEは、分岐動作を所定のサイクルで開始すべき場合にしばしば要求される。

シーケンサSEQ は、バスライン(PPC、図3参照)を形成する記憶回路30の入力部に記憶される疑似プログラム計数値PPCも発生させる。PPCを、分岐ユニッ

トBRを含む他の機能ユニットによって使用して、プログラム計数-相関アドレス指定を実行することができる。このために、読出し遅延を、バスラインから記憶ユニットの値PPCを読み出す命令のサイクル数とPPCの値に等しいPCを有する命令のサイクル数との間の差として、規定することができる。

動作中、bサイクルの分岐遅延が、分岐命令を発生させるサイクルiと、命令ワードIW(i+b)を分岐命令によって行う第1サイクルi+bとの間で発生する。分岐命令の条件の同一ブール値を、ガードビット入力部AL.G, RG.G, DM.G及びBR.Gに指導することができ、かつ、別の分岐を含む、分岐遅延中の種々の機能ユニットに対してスケジュールされた状態変化動作を条件的に禁止する調停サイクルに使用することができる。分岐の可能性を知ることにより、起こりうる条件的な分岐命令の分岐遅延内で、有効な動作をコンパイラによってスケジュールすることができ、同時に、分岐条件が適合する場合の少ないときに正確な状態を満足する。ガードされていない動作を最も都合良く開始するために、各ガードビット記憶ユニットの書込み不可能な位置を、適切な読出しアドレス.R.Aに応答してその読出しポートに真を発生させるように配置する。32ビットのオペランド記憶ユニットの書込み不可能な位置を、ある既知の、頻繁に要求される値、例えば零を発生させるように配置することもできる。

オフチップメモリ、レジスタユニットRGのレジスタ及びプログラムカウンタPCの内容のみを、本形態の処理装置のプログラマ可視状態の部分と考える。イミテーションマルチポート記憶回路30を、厳密に一時的な中間変数に対するバッファとしてのコンパイラによって使用することができる。その結果、コンスタント

(18)

特表平9-509515

ユニットCOのような機能ユニットは、ガードのために必要とされない。計算及び論理ユニットALそれ自体が装置のプログラマ可視状態を変更することができなくても、ユニットALに、疑似的に生じる計算のオーバーフローのような例外事態を回避するガードを設ける。

分岐ユニットBRの分岐遅延に加えて、スケジューリングコンパイラは、各機能ユニットのいわゆる「ラテンシー」の知識を有する必要がある。機能ユニットのラテンシーを、動作を開始するサイクルの数と、その動作の結果(AL.0等)を発生させるとともに宛先制御フィールドDECによりイミテーションマルチポート記

憶回路30の一つ又は複数の所望の位置にその結果を指導するサイクルの数との間の差として規定する。既に説明した分岐遅延を、分岐ユニットBRが原因のラテンシーとシーケンサSEQ及びオフチップ命令発生パイプラインIIPが原因のラテンシーとの和と考えることができる。

スケジューリング例

以下の例のために、機能ユニットに対するラテンシーを次のように仮定する。

ユニットCO—零サイクルラテンシー

ユニットAL—1サイクルラテンシー

ユニットRG—1サイクルラテンシー

ユニットDM—2サイクルラテンシー

ユニットBR—5サイクル分岐遅延

下記のリスト1は、ハイレベル言語ソースコードのフラグメントを具える。メモリ中の二つの記録を、変数p及びqで示す。プログラムの目的は、二つの32ビットの値を、q記録のフィールドから二つの対応するp記録のフィールドにコピーすることである。

リスト1

p[^]. フィールド1 := q[^]. フィールド1 ;

p[^]. フィールド2 := q[^]. フィールド2 ;

図4は、リスト1のハイレベル命令を実行するのに使用することができるデータのあり得る構成を図示する。プログラム変数の記憶を、0から15の番号を付し

(19)

特表平9-509515

た16個の位置を有する高速レジスタファイルREG及び(オフチップ)読出し/書込みメモリのより大きなバンクMEMによって行う。ローカルベースアドレスLBを、レジスタファイルREGの位置0に記憶させる。アドレスLBは、メモリバンクMEMの可変値のテーブルの開始を示し、この場合、二つの記録ポインタp及びqは、位置LB+12及びLB+8にそれぞれ記録される。各変数p及びqを、メモリバンクMEM以外に配置された、対応する記録の開始に対するポインタとする。各記録の第1フィールドを、関連のポインタ変数によって規定されたアドレスから零のオフセットに記憶させる。各記録の第2フィールドを、関連のポインタ変数から4のオフセットに記憶させる。したがって、値q、フィールド2を、アドレスがq+4である位置から読み出すことができる。

リスト2は、図4を参照して説明したデータ構成を仮定した場合、VLIWプロセッサでリスト1のハイレベル命令を実行するコンパイラによって発生させるT1を付した中間コードフラグメントを示す。リスト2の各行は、適切な機能ユニットによる実行に対してスケジュールすべき単一基本動作Op001 ~ Op014を規定する。各行の終わりのコメント(、...,、)は、最初に、要求される機能ユニットのタイプ(AL, RG, CO等)を言及し、その後、その動作の結果(出力)を説明する。

(例えば)動作Op003の参照番号001及び002は、Op003に対するオペランドがそれぞれOp001の結果及びOp002の結果であることを表す。したがって、動作Op003の影響を、動作Op001の結果及び動作Op002の結果に加算し、同時に、動作Op009の影響により、Op008の結果を、オフチップデータメモリのOp004の結果によってアドレス指定された位置に書き込むべきである。この情報に上記図4の説明を組み合わせると、リスト1で特定した機能を実行するリスト2の各動作Op001 ~ Op014の目的は自明となる。

リスト2

T1:

```
Op001  rgreg(0);      (、RG,ローカルベースアドレス、)
Op002  constant(12);  (、CO,pに対するオフセット値、)
Op003  aplus 001,002; (、AL,pのアドレス、)
```

(20)

特表平9-509515

```

Op004 read 003;      (* DM,p *)
Op005 constant(8)    (* CO,q に対するオフセット値 *)
Op006 aplus 001,005; (* AL,q のアドレス *)
Op007 read 006;      (* DM,q *)
Op008 read 007;      (* DM,q ^, フィールド1 *)
Op009 write 004 008;  (* DM,p ^, フィールド1:=q ^, フィールド1 *)
Op010 constant(4);   (* CO, フィールド2 に対するオフセット *)
Op011 aplus 004,010; (* AL,p ^, フィールド2 のアドレス *)
Op012 aplus 007,010; (* AL,q ^, フィールド2 のアドレス *)
Op013 read 012;      (* DM,q ^, フィールド2 *)

Op014 write 011,013; (* DM,p ^, フィールド2:=q ^, フィールド2 *)
GOTO EXIT

```

中間コードフラグメントT1が順次リストされ、少なくとも一つの機能ユニットが各動作を実行するのに適切である場合には、このフラグメントがVLIWプロセッサの任意の特定形態に特定されないことがわかる。一旦、コンパイラが、アプリケーションプログラムを構成するこのようなフラグメントの収集に到達し、かつ、一旦、実際の装置形態が既知となると、フラグメントの各動作を特定サイクルの装置の特定の機能ユニットに写像することは、スケジューラのタスクである。スケジューラは、特定のVLIWプロセッサの構成によって規定された一組の制約によって動作する。これらの制約は、ターゲットVLIWプロセッサで利用できる機能ユニットの数及びタイプと、各機能ユニットのラテンシーとを主に含む。他の制約は、イミテーションマルチポートメモリの各記憶ユニットの位置の数を含む。これらの位置は、スケジューラにより必要に応じて動的に割り当てられる。フラグメントT1によって表される簡単なスケジューリングの問題のために、図1～4のVLIW処理装置の各記憶ユニットの位置の数が、記憶ユニットの容量の制限を本明細書の制約とする必要がない程度に十分であることがわかる。

他の動作の結果に依存する動作を、全ての他の動作のラテンシーを満了させるまでスケジュールする必要がないので、各動作のラテンシーは重要である。した

(21)

特表平9-509515

がって、例えば、動作Op009を、(DMラテンシーが2である)Op004より後で少なくとも2サイクル及びOp008より後で少なくとも2サイクルスケジュールする必要がある。同様に、動作Op011を、(COラテンシーが零である)Op010と同一サイクルでスケジュールすることができるが、(DMラテンシーが2である)Op004より少なくとも2サイクル後とする必要がある。これは、記憶ユニット位置の書込みと読出しとを同時に行えることを仮定する。そうでない場合、各タイプの動作のために、別の1サイクルのラテンシーを必要とするおそれがある。

図5は、各々が中間コードフラグメントT1を、図1～4を参照して説明した特定のVLIWプロセッサに写像する、三つの相違するスケジュール(I,II及びIII)を示す。各表の各行は、一つの非常に長い命令ワードIWしたがって一つのマシンサイクルを表す。Cyを頭につけたサイクル番号を、最も左の欄に付す。次の五つの

欄は、五つの機能ユニット(図1のCO等)に対する五つの制御フィールドCOC,ALC,RGC,DMC及びBRCを表す。各制御フィールドは、各機能ユニットで動作を開始する。したがって、スケジュールIでは、サイクル0で、フィールドCOCが、コンスタントユニット(図1のCO)に対して、Op005を開始するように命令し、その結果、結果CO.0=8が(COラテンシーが零である)同一サイクルのユニットCOの出力部に現れる。同様に、レジスタの読出し動作Op001を、サイクル0でフィールドRGCによって開始する。ローカルベースアドレスの値LBはしたがって、(RGラテンシーが1である)サイクル1でRG.0である。

各命令中の宛先制御フィールドDECの選択フィールドDEC.—.SEを、機能ユニット制御フィールドの右に表す。これらフィールドは、種々の動作Op001～Op014の結果を、このような結果を利用できるように各記憶ユニットに指導する。スケジューラは、規定した宛先制御フィールドDECの種々の機能ユニットのラテンシーを考慮する。したがって、例えば、サイクル0のフィールドRGCから開始したレジスタ読出し動作Op001の結果RG.0は、次の命令、サイクル1のフィールドDEC.AL.L.SEにより、計算及び論理ユニットALの左側オペランド記憶ユニットに指導される。これを、スケジュールIのサイクル1のAL.L列のコードRGで表す。同様に、メモリ読出し動作Op007をサイクル2のフィールドDMCから開始し、

(22)

特表平9-509515

同時に、その動作の結果DM.0を、(DMラテンシーが2である) サイクル4でフィールドDEC.DM.A.SE に 'DM' を設定することにより、データ記憶ユニットDMのアドレス記憶ユニットに指導する。

スケジューラは、それ自身の動作Nw015 及びNw016 を加算して、次のフラグメントへの飛び越しを行う(リスト2の 'GOTO EXIT')。Nw015 は、サイクル1でコンスタントユニットCOを用いて、サイクル1で選択フィールドDEC.BR.A.SE を 'CO' に設定することにより分岐ユニットBRのアドレス記憶ユニットに指導される分岐に対するターゲットアドレスを規定する。Nw016 は、サイクル1で分岐制御フィールドBRC を用いて、ターゲットアドレスに対する無条件(BR.G=1)分岐を開始する。5サイクルの長い分岐遅延により、フラグメントT1完了後、ターゲットアドレスに対する制御の転送がサイクル8まで生じなくなる。換言すれば、分岐動作は、スケジュール中に提示されて、有効な動作を、分岐遅延中の実行に對し

てスケジュールすることができる。起こりうる最大数の動作の約1/3のみが図5のスケジュールIで実際にスケジュールされる。この原因は、非常に小さいサイズのフラグメントの例であるからであり、フラグメントが長くなるに従って一致の程度が高くなるので、より長いフラグメントをスケジュールすると、割合の高い有効な動作をスケジュールすることができるのは、当業者には理解することができる。

より長いフラグメントに固有の多数の条件的な分岐動作は、1又は0に近接する分岐確率を有するように大抵の条件的な分岐を配置することができることを利用しうるブール値のガードビット.Gの利用により説明した実施の形態で克服される。例えば、100回実行すべきループは、0.99又は0.01の確率で生じると予測できる分岐を意味する。

図5のスケジュールIを、真のマルチポートメモリを有するVLIWマシンによって実行し、宛先制御フィールドDEC を完全に無視した。しかしながら、スケジュールIは、イミテーションマルチポートメモリの記憶ユニットに対するアクセスの衝突を含み、したがって、図1～4のプロセッサに対する動作Op001 ~Op014

(23)

特表平9-509515

の有効な写像ではない。特に、スケジュールⅠのサイクル5では、Op004の結果DM.0は、計算及び論理ユニットALの左側オペランド記憶ユニット(AL.L)及びデータ記憶ユニットDMのアドレス記憶ユニット(DM.A)に送られる。同一サイクル4において、Op012の結果AL.0も、データ記憶ユニットDMのアドレス記憶ユニット(DM.A)に送られる。この衝突は、スケジュールⅠのサイクル5においてフィールドDEC.DM.A.SEのコードXXによってマークされる。

遅延素子(特定レジスタSP)が存在しない場合、スケジューラは、プロセッサハードウェアの制約が満足されるまで、衝突を発見する際これら動作を再スケジュールするように強制される。本例では、これは図5のスケジュールⅡとなる。スケジュールⅡにおいて、Op004は、1サイクル遅延されて、サイクル4で開始する。しかしながら、動作Op009,Op011及びOp014はOp004の結果に依存し、したがって、これら動作も少なくとも1サイクル遅延する。これら動作は、スケジュールⅠにおいて、他の動作Op012,Op008,Op013及びOp014と互いに重なり合い、ハードウェアの制約によりこれら動作も再スケジュールする必要がある。

この再スケジュールのプロセス、すなわち「後戻り動作」を、常にハードウェア制約を満足するように完了させることができるが、再スケジュールされたフラグメントⅡはこの場合、8の代わりに9サイクル占有する。これは、同一機能ユニットを有する理想的なVLIWマシンの場合に比べてアプリケーションプログラムのこの部分を長く取ることを意味する。さらに、後戻り動作は、最初の衝突を解決する際に発生する新たな衝突の解決を含む、追加のワーク中のコンパイラを意味する。増大するコンパイラの影響の問題は、衝突がほとんど毎回のサイクルで発生する大きなマシンで特に重大となる。衝突が発生したサイクルを超えて複数の命令がスケジュールされるまで、衝突の存在を一般に検出できないことは明らかである。10又はそれ以上の機能ユニットの大きなマシンでは、これは、再スケジュールリング中頻繁なものとして生じる新たな衝突が起こりうる、数十回又は数百回の動作の再スケジュールリングを要求する。

図5のスケジュールⅢは、同時継続出願594,534号により詳しく説明したような特定レジスタSPの形態の遅延素子を用いる、後戻り動作のない衝突の例の分

(24)

特表平9-509515

析を示す。宛先選択フィールドDECの選択フィールドDEC.SP.SEを、スケジュールIIIの右側に示す。動作Op001 ~ Op014 及びNw015 ~ Nw016 のスケジュールリングはスケジュールI（精密なマルチポートメモリ）のものとはほぼ同一であることがわかる。しかしながら、スケジュールIIIにおいて、DM.A記憶ユニットに行く二つの結果間のサイクル5の衝突は、Op004の結果DM.Oを特定レジスタに指導することにより、すなわちフィールドDEC.SP.SEを宛先制御フィールドDECの‘DM’に設定することにより、解決することができる。フィールドDEC.DM.A.SEはこの際、サイクル6で‘SP’に設定されて、サイクル6のOp009でデータ記憶ユニットDMによって読み出すべきときにOp004の遅延された結果をDM.A記憶ユニットに指導する。

Op004の遅延した結果はこの際、サイクル6においてOp011の結果と衝突し、これら結果は双方ともDM.A記憶ユニットに行く。したがって、スケジュールIIIにおいて、Op011の開始は、他の任意の動作に連続的な影響を及ぼすことなくサイクル6まで遅延される。この第2の衝突に対する他の解決は、サイクル5においてOp011をそのままにし、同時に、サイクル6においてフィールドDEC.SP.SE

を‘AL’に設定することによりOp011の結果を遅延させることである。Op011の遅延した結果は、サイクル7でフィールドDEC.DM.A.SEを‘SP’に設定することによりOp014の開始時にDM.A記憶ユニットに指導することができる。

特定レジスタSPは常に、単一の、二つの値の衝突が任意の所定のサイクルで生じる間、後戻り動作を回避しながら其のマルチポートマシンの性能レベルを維持するようにしている。これは、機能ユニットが各サイクルで記憶ユニットから一つの値のみしか読み出すことができないということになり、衝突結果の一方が常に他方の前に要求されることを意味する。論議のために、Op011の結果がOp004の結果に比べてより要求されるものであることがわかる場合、Op004の結果は、それが衝突結果より緊急なものとなるまで、2サイクルすなわち任意の数のサイクル遅延させることができる。全ての場合において、他の動作の連続的な再スケジュールリングの必要は回避される。

包含すべき機能ユニットの数及びタイプの選択を十分自由に利用することがで

(25)

特表平9-509515

きる。図1の特定の形態の各ユニットが、特定機能又は機能の群を実行するのにある程度特定される間、これら機能の群を全て、スケジューリングにおける最大柔軟性に対して、より一般的な多目的タイプの機能ユニットに組み合わせることができる。このタイプのユニットの不都合は、十分なパイプラインを許容するために、所定のユニットの全ての動作に対してある程度のラテンシーを有することである。これは、(一定発生のような)より迅速な動作を、最低動作速度までスローダウンする必要があることを意味する。それに対して、より特定された機能ユニットタイプを、例えば浮遊点乗算機能、バレルシフト機能又は入出力機能を実行するために、所定の用途でも利用できるようにする必要がある。通常のマイクロプロセッサで用いられるのと同タイプのユニットを、本発明の処理装置に含有するために当業者によって良好に配置すなわち容易に適合させる。

例外事態回復を有するメモリインタフェース回路

図6は、図1～4の処理装置のデータメモリユニットDMとして使用するのに適切なメモリインタフェース回路の構成を示す。信号DM.G(ガードビット)、DM.A(アドレス)及びDM.I(言込みデータ)を、イミテーションマルチポイントメモ

リ30(図1参照)から受信する。記憶管理ユニット(MMU)は、(図1に図示しない)ある例外事態回復回路100を介してこれら信号を受信するとともに、信号DM.Oを、回路100を介して記憶回路30に供給する。OPコード信号DM.OP(読出し/書込み)を、命令レジスタ46(図1参照)から受信する。MMUは、大規模なオフチップ主メモリMBMに対するインタフェースを提供し、特に、それ自体は当業者によく知られた仮想記憶アドレス指定を実行する。

例外事態回復回路100内では、信号DM.A及びDM.Iには影響が及ぼされず、同時に、保護ビットDM.Gは、信号DM.G'としてMMUに通過させる前に、保護強制信号GFとともにAND演算される。信号GFが論理'1'(真)の状態である場合、信号DM.G'は信号DM.Gに等しい。しかしながら、信号GFが論理'0'(偽)である場合、DM.G'は'0'(偽)に強えられる。保護強制信号GFは、図1の処理装置の他の機能ユニットの保護ビットAL.G, RG.G及びBR.Gとともに、(説明しない)

(26)

特表平9-509515

同様な方法でAND演算される。

例外事態回復回路100は、マルチプレクサ102及び記録／再生回路RRCも含む。回路RRCは、再生モード信号REPによってマルチプレクサ102を制御する。REPが論理‘0’の間、マルチプレクサは、MMUから発生した信号DM.O'を、信号DM.Oとしてイミテーションマルチポート記憶回路30に送信する。しかしながら、REPが論理‘1’の間、記録／再生回路RRCから発生した再生データ信号DREPは、DM.O'の代わりに記憶回路30に送信される。したがって、再生モード信号REPは、DM.O=DM.O'である記録モードを、DM.O=DREPである再生モードから区別する。記録／再生回路RRCも、マルチプレクサ102の出力部から信号DREC=DM.Oを取り出す。

(信号GF=‘1’及びREP=‘0’によって特徴付けられた)通常動作では、破線ボックス100は実質的には透明となり、その結果DM.G'=DM.G及びDM.O=DM.O'となる。MMUは、各サイクルで、主メモリMEMの位置DM.Aから値DM.Oを読み出す、又は、opコードDM.OPに依存するとともに条件DM.C=真に支配されて、主メモリMEMの位置DM.Aに値DM.Iを書き込む要求を受け取る。MMUは、記録／再生回路RRCにメモリMEMから読み出された値DM.Oの可用性を表す事象信号ETV(ETV=1)を、記録／再生回路RRCに供給する。

MMUはまた、後に説明するTLBロス信号TLBMをORゲート104の入力部に供給し、そのORゲートの出力部は、例外事態検出信号EXDETを、シーケンサSEQ及び命令発生パイプラインIIP(図1参照)に搬送する。ORゲート104の他方の入力部を、他の機能ユニットから例外事象立上がり信号を受信するのに利用する。信号EXDETを、マシンサイクルごとに一度同期されるセッターリセットフリップフロップ106のリセット入力部(R)にも供給する。命令発生パイプラインIIPは、例外事態ハンドリング信号EXH及びチェックポイント信号CPTを発生させ、かつ、これら信号を記録／再生回路RRCに供給する。例外事態ハンドリング信号EXHを、フリップフロップ106のセット入力部(S)にも供給し、同時に、フリップフロップ106の出力信号(Q)は、保護強制信号GFを構成する。

図7は、記録／再生回路RRCをより詳細に示す。マシンサイクルごとに一度同

(27)

特表平9-509515

期される時間カウンタTICは、チェックポイント信号CPTを受信し、かつ、時間信号TRECを発生させる。時間信号TRECを、信号DRECに連結して、事象メモリEQMのデータ入力部(DIN)に供給する。信号DM.0は32ビットを有するので、時間信号TRECに対して8ビットの幅と仮定すると、事象メモリEQMのデータ入力幅は40ビットの幅を有する。時間信号TRECはまた、チェックポイント信号CPT、事象信号EVT及び例外事態ハンドリング信号EXHのように、事象カウンタ論理回路EVCに供給される。論理回路EVCは、アドレス信号ADDR及び書き込みイネーブル信号WEの各々によって事象メモリEQMを制御する。

事象メモリEQMは、40ビットのデータ出力DOUTを有し、この出力のうちの8ビットの事象時間信号TREP及び32ビットの再生データ信号を、論理回路EVC及びラッチ108にそれぞれ供給する。ラッチ108は、論理回路EVCから供給される次の事象信号NXTによって同期され、同時に、ラッチ出力部は、再生データ信号EREPをマルチプレクサ102に供給する。

当業者には十分既知の仮想記憶動作に対する記憶管理ユニットの構成及び動作の詳細な説明は、本発明の理解のためにここでは必要でない。要約すれば、本形態のMMUは変換索引バッファ(TB又はTLB)を含み、この変換索引バッファは、仮想アドレス(DM.A)をメモリMBMの物理アドレスに写像する複数のエントリを含む。あり得るアドレスのサブセットのみをTLBにより所定の時間に写像することが

でき、TLBに存在するエントリによってアドレスDM.Aを写像することができる場合、アドレスDM.Aに対する読出し又は書き込みの要求を、データメモリユニットラテンシー内(本形態の2サイクル)内でのみ適合させることができる。アドレスDM.Aが、存在するエントリによって写像できない場合、これは例外事態を構成し、プログラムの実行を中断する必要がある。このいわゆるTLBロスを、(存在するエントリを削除する)新たなTLBエントリを作成することにより解決することができるので、TLBロスを、「非重要」又は「回復可能」例外事態と称する。MMUにより信号TLBM=1を用いて、TLBロスの発生を表すとともに、例外事態ハンドリング及び回復プロセスを開始する。

例外事態を有する実行例

(28)

特表平9-509515

図8は、複数の非常に長い命令ワード（図2のIW参照）に既にスケジュールされ、プログラムカウンタ位置 $PC=0,1,2$ 以下参照に記憶された第2例のプログラムフラグメントを線形的に示す。全てのスケジュールされた動作を図示するわけではなく、実行の際に主メモリMEMの所定の位置に記憶された値に1増分する五つの動作を示す。フラグメントは、イミテーションマルチポート記憶回路30に既に存在する値を故意に利用しない。したがって、 $PC=0$ における関連のマシン状態は、完全にデータメモリユニットDM（主メモリMEM）及びレジスタユニットRGのレジスタに存在する。さらに、 $PC=0$ の前に完了しない動作は、 $PC=0$ の前でスケジュールされない。したがって、 $PC=0$ に、図8においてアスタリスク「*」によって表したチェックポイント命令を示すことができる。制御フローがチェックポイントで「分岐する」場合、この後者の状態は常に必要でなく、これは、チェックポイント到達時に中間ヒストリ（パイプラインの内容）が常に同一であることを意味する。

第2例における $PC=0$ のフラグメントにエントリすると、関連の記憶位置に対するポインタは、レジスタユニットRGのレジスタ0に記憶される。 $PC=0$ で開始される動作'rdreg'は、ここからのポインタを、イミテーションマルチポート記憶回路30の一時的な位置t1にコピーする。この場合、t1を（RGラテンシーが1である） $PC=1$ で受け取る。 $PC=1$ で開始される動作'read t1'は、関連の記憶位置に記憶された値を第2の一時的な位置t2で読み出す。この場合、t2を（DMラテンシーが

2である） $PC=3$ で受け取る。その間、 $PC=2$ で開始される動作'constant(1)'は、増分値1を、（COラテンシーが0である）第3の一時的な位置t3にロードする。 $PC=3$ で開始される動作'apltus t2 t3'は、メモリt2及び増分t3からの値の和を、第4の一時的な位置t4に置換する。この場合、t4を、（ALラテンシーが1である） $PC=4$ で受け取る。最後に、 $PC=4$ で開始する動作'write t1 t4'は、一時的な位置t1も表した主記憶位置に戻って増分した値t4を書き込む。この場合、t4は、（TLBロスがないと仮定して） $PC=6$ に到達する。

波形図9は、図8の第2例のプログラムフラグメントを $PC=5$ で実行する際に例外事態（例えば、TBL ロス）を検出すると仮定する場合の、図6及び7の回路を

(29)

特表平9-509515

用いた例外事態回復プロセスを示す。図9の最上行'PC='は、チェックポイントPC=0に対する分岐を有するPC=Bで開始する、処理装置の連続的なサイクルにおいてIIPパイプラインから発生する構成のプログラム計数値を表す。以下、信号CPT、(時間信号計数値を示す) TREC, EVT, EXDET, GF, EXH 及びREP の波形を表す。波形の開始時において、信号CPT, EVT, EXDET, EXH 及びREP を全て'0'とし、GFを'1'とする。これらは、通常動作(例外事態を取り扱わない記録モード)を表す。時間計数信号TRECは、(図示しない)以前のチェックポイントであるので計数サイクルとなる。

チェックポイント命令を、関連の命令ワード(図2参照)にビットCP='1'を設定することにより、プログラムメモリを表す。本形態では、ビットCPを、チェックポイント命令それ自体に付属するというよりはチェックポイントに対する分岐動作を開始する命令PC=Bに設定する。この手段は、5サイクルの各チェックポイントの有利な警告を付与し、分岐遅延により、緩和タイミングは、チェックポイント信号CPT=1 を発生させ又は別の方法でチェックポイントに応答する回路を制約する。当然、分岐動作がガードされるFALSE(BR.G=FALSE及び/又はGF=0)である場合には信号CPT=1 は発生しない。分岐遅延の最後のサイクルPC=Lの最後に対して、チェックポイント信号CPT を、'C'を付したサイクルの開始時に、命令発生パイプラインによって(論理「1」に)設定する。この場合、最初のときにはPC=0とする。

例外事態に続くようにリターンする必要がある場合、チェックポイント信号CP

T=1 により、シーケンサSEQ にチェックポイントのPC値を記憶させる。また、チェックポイント信号CPT=1 により、記録/再生回路RRC(図7)内の時間カウンタTIC を零にリセットし(TREC=0)、事象計数論理回路EVC によりアドレスADDRを零に設定し、事象メモリEVM を有効にクリアする。時間計数信号TRECを次のサイクルでそれぞれ増分して、チェックポイントからの時間経過の目安として作用する。事象信号EVT を、メモリから読み出された値を(信号DM.0=DM.0'の形態で)最初に利用できる際、Tを付した(PC=3)サイクルに対するとときにMMU によって設定する。事象信号ETV=1 に応答して、事象計数論理回路EVC は、事象メモリEVM

(30)

特表平9-509515

の言込みイネーブル信号WEを設定して、記録データ信号DREC=DM.0=DM.0'を、時間計数信号TRECの形態の「時間スタンプ」とともに、現在3に等しい事象メモリEVMの最初の位置に記憶させる。これに従って、事象メモリアドレスADDRを、次の事象の記憶の準備のために増分する。

E(PC=5)において、例外事態検出信号EXDET=1によって示したように、TLBロス又は他の例外事態が発生する。これにより、フリップフロップ106にガード強制信号GF=0が発生させ、その間PC=6及びPC=7の実行を継続する。これら二つのサイクルにより、例えば動作'write t1'を含む任意の動作を例外事態前に開始して、それらの実行を完了する。したがって、例外事態が検出される前に開始されるこれら及び他の動作は、マシン状態を、チェックポイントで存在する良好に規定された状態から変更する。これら二つのサイクル中に新たな動作を開始しても、保護強制信号GF=0により、例外事態が生じたのちに開始される動作はマシン状態に影響を及ぼさず、その結果、マシン状態は時間Rで正確である。

EとRとの間の2サイクル遅延はシーケンサ及び命令発生パイプラインによっても用いられて、TLBロスの例外事態の場合、MMUのTLBの新たなエントリをセットアップする例外事態ハンドリングルーチン(PC=X)に対する転送制御を行う。この遅延は通常の分岐遅延より短く、例外事態の処理に対する時間の不利益を最小にする。例外事態処理ルーチンの持続時間を、ガード強制信号GFを'1'にリターンすることに対応して、時間Rで発生する信号EX=1によりマークされ、時間カウンタTICは計数を停止する(TREC=7)。

例外事態処理ルーチンは、永久にかつ直接にアドレス指定可能なメモリの小区

域を用いて操作される。この手段及び適切なプログラミングにより、例外事態処理ルーチンそれ自体が重要でない例外事態を絶対発生させないことを保証する。この手段はそれ自体、例えば既に説明した文献Colwell等から当業者には既知である。例外事態処理ルーチンの要求される動作は、当業者には十分既知の方法で、MMUに依存し、これらの他の記載は、ここに記載される例外事態回復処理の理解に必要なでない。例外事態ハンドリングも、プログラマにより明らかに準備することなく、特定のハードウェアによって実行することができる。例えば680X0プ

(31)

特表平9-509515

ロセッサにおいて、特定の「テーブルウォーク」ハードウェアを、TLB 損失の例外事態を処理するために設ける。

例外事態処理ルーチンの終了に向かって、シーケンサ SEQ は最近のチェックポイント（本例では $PC=0$ ）の PC 値を回復し、それを命令発生パイプライン IIP に入力し、その結果、 $PC=0$ で記憶された命令ワード IW を、図 9 で C' を付した時間に命令レジスタ 46 にロードし、これにより、チェックポイント $PC=0$ から開始する第 2 例のプログラムフラグメントの再実行を開始する。チェックポイント信号 CPT を、記録／再生回路 RRC の時間カウンタ TIC が時間信号 $TREC$ を零にリセットするのに応答してこのサイクルのときに設定し、このチェックポイント信号により、チェックポイントからのサイクルの計数も開始する。事象計数論理回路 ECV は事象メモリのアドレス $ADDR$ を零にリセットし、その結果 T で記録された値 $DREC$ 及び時間スタンプ $TREP=3$ は、事象メモリ EVM のデータ出力部 $DOUT$ で利用できる。

再実行に当たり、事象計数論理回路 EVC は再生選択信号 $REP=1$ を設定する。したがって、マルチプレクサ 102 は、MMU から発生した信号 $DM.0'$ のデータ経路を中断し、ラッチ 108 を介して事象メモリ EVM から発生した再生データ信号 $DR EP$ を構成する。再実行の時間 T' において、回路 EVC は、現在の時間信号 $TREC$ と記憶された時間スタンプ $TREP$ との間が等しいことを検出し、信号 NXT を用いて、本来の実行中の時間 E に記録された値をラッチ 108 に入力する。したがって、時間 T' でイミテーションマルチポート記憶回路 30 に供給された値 $DM.0$ は、本来の時間 T のメモリ MEM の値となり、プログラムフラグメントの本来の実行中の動作 'write t1 t4' によってここに記憶された増分値とならない。

したがって、再生モード中、実行により悪く規定された実際のマシン状態に依

存する値の代わりに、以前の正確なマシン状態に依存する値が事象メモリ EVM から供給されることがわかる。再実行中の $PC=6$ において、動作 'write t1 t4' は単に、本来の実行中メモリ MEM に記憶された増分値に同一値を重ね書きするだけである。正確な入力のみが（あたかも）再実行中のマシン状態から受信されるので、マシン状態に対する正確な出力が保証される。実際には、以下の個別の見出しで説明する理由により、全ての読出し及び回路動作を再生モード中に行わないの

(32)

特表平9-509515

が好適である。

図8及び9の実行例に戻ると、回路EVCは、例外事態を処理するために実行が中断された点を超えて再実行される時間 R' ($PC=8$)の直前に再生信号REPをリセットする。これにより、任意の他の事象(データ転送)を、事象メモリEVMに以前に記録された(事象Iまでの)事象に追加し、したがって、同一チェックポイントから再実行を要求することができる任意の一つ又は複数の他の事象の場合に正確な回復が可能となる。

より大規模な処理装置の場合、複数のデータ記憶ユニットDMを設けて、各サイクルにおける主メモリMEMに対する多重アクセスを可能にする。当然、このような実施の形態では、記録/再生回路には主メモリMEMからの各データ経路が要求される。全体の処理装置が同期して作動するので、回路RRCのある素子、例えば時間カウンタTICを、二つ又はそれ以上の回路間で共有することができる。これら他の記憶ユニットは、並列処理の範囲を増大させるために、他の個別のメモリより同一主メモリスペースMEMに追加のパーツを理想的に提供する。

レジスタユニットRGはプログラマ可視マシン状態も含むので、例外事態回復の特徴も設ける。しかしながら、ユニットRGの状態の量が非常に小さいので、バックアップコピーレジスタを用いて、既に引用された文献のHwu及びpattによる例に記載したような通常の方法で、各チェックポイントにおける状態を記憶覆る。したがって、レジスタユニットの例外事態回復特徴を、簡潔のために更に記載しない。代案として、レジスタの数が完全なバックアップコピーに対して非常に多くて経済的に収容できない場合、(一つ又は各)レジスタユニット出力データ経路(RG.O)が記録/再生回路を含むことができることは、当業者にはわかる。

更に別の実施の形態において、イミテーションマルチポートメモリの内容それ

自体をマシン状態の定義に含めることができ、この場合、各メモリユニット(26, 図1)は、当該記憶ユニットのサイズに応じて、バックアップ記憶ユニット又は記録/再生回路の形態で例外事態回復特徴を設ける。

時間スタンピングのない記録/再生回路

図10は、図7の記録/再生回路の他の例を示し、この場合、時間カウンタTI

(33)

特表平9-509515

C を設けず、時間スタンプ値(TREC)を事象メモリEVM のデータエントリを用いて記憶させない。これは、明らかに、事象メモリのサイズを減少させるとともにその周辺の制御回路を簡単化するが、チェックポイント命令の最初の実行中の時間Tに行ったような再生中の時間T'に(MMU-図6 から生じた) 事象信号EVT が発生するということに依存する。したがって、この他の例において、記録/再生回路は、記録モード中事象信号EVT が表明されると常に新たな値DRECを記憶するとともに再生モード中事象信号が表明されると新たに記憶された値を再生する「先き入れ」「先き出し」すなわちFIFOメモリを有効に提供する。

当然、事象記憶制御回路EVC は、チェックポイントから正確に記録された値の数の計数を維持して、アドレスADDRを発生させるとともに、以前の試みで実行が中断した点を再実行が通過する際に再生モードから記録モードへの切替を行う。当然、図10の簡単な例に対して、システム設計者は、アドレスADDRの開始値及び事象信号EVT の表明の数が記録モードと再生モードの両方で正確に対応する、すなわち再生値はもはや正確に記録した値を再生しないことを満足させる必要がある。

再生モードのメモリアクセスの制約

上記実施の形態の変形を実際のシステムに要求することができる。例えば、既に説明したように、第2例のプログラムフラグメント(図8及び9)の再実行において命令't2←read t1'を再実行し、その後命令'write t1 t4'を許容して、本来の実行中同一命令により書き込まれた値を処理し及び重ね書きする。アドレスt1から読み出された不正確な値が無視される間、及び、アドレスt1に重ね書きされた値が正確である間、一般に再生モード中の正確な読出し動作及び書き込み動作を禁止するのが望ましい。

この理由は、記憶位置のアクセス動作は他の事象をトリガすることがしばしば

あり、したがって不所望な繰り返しが行われる。実際には、シリアル通信のUARTのような入出力装置はしばしば単なる記憶位置(「マップされたメモリ」としてアドレス指定される。しかしながら、このような場合、記憶位置に同一値を2回書き込む動作により、値が遠隔装置に2回送信される。文字列'DATA'を受信

(34)

特表平9-509515

する代わりに、遠隔装置は'DAATA'を受信する。同様に、UARTレジスタから受信した文字の読出し動作は通常、ワードがプロセッサによって許可されるUARTに信号送信されて、読出しの繰り返しにより、同一でない次の文字が生じる。したがって、遠隔装置が列'DATA'を送信する場合、プロセッサは'DAA'のみを受信する。

メモリマップしたI/O装置が存在しない場合でさえ、繰り返しメモリ読出し動作は、仮想記憶システムの予期しない事象をトリガすることができる。例えば、第2例のプログラムフラグメントの命令't2←read t1'の実行に当たり、記憶位置t1を正確にアクセスするとともにデータ変換が事象メモリEVMに記録されたと仮定する。さらに、例外事態を処理する間、位置t1を含むメモリのページが、TLBの新たなページ参照を詰めるために捨てられると仮定する。再実行がチェックポイントから始まると、要求される値が事象メモリEVMで記録されても、最終時間周辺で成功した記憶読出し動作を実行するとTLBロスの例外事態が生じる。明らかに、この状況は不所望であり、この理由に対しても、メモリアクセスを再生モード中有効に抑制するように配置することは極めて望ましい。

制御パイプラインの削除

簡単なラテンシー値より詳細な動作の各タイプのタイミングの知識をコンパイラに付与することにより、設計者は場合によっては「制御パイプライン」を削除し又は減少させることができるのがわかる。この技術により処理装置から多数の遅延回路を削除し、これにより物理的な回路寸法を減少させる。代わりに、コンパイラがその詳細な知識を用いて、所望の動作に関連する相違したビットフィールドを個別にスケジュールし、その結果、各ビットすなわちビットフィールドは命令レジスタ46にロードされず、実際に要求されるマシンサイクルまで機能回路に提供されない。

例えば、所定のサイクル($PC=i$)において、4サイクルのラテンシーを有する機能ユニットによって実行すべき動作を開始したいと仮定する。ここで説明する処理装置の実例では、ユニットに対するガード選択フィールド(—G.RA)を $PC=i+OD$ 発生させることができ、opコードを $PC=i+1$ で発生させることができ、一つ又

(35)

特表平9-509515

はそれ以上のアーギュメント選択フィールド(—, —, RA)を、 $PC=i+2$ 又は $PC=i+3$ で発生させることができる。宛先制御フィールド(DEC, —, —, SE及びDEC, —, —, WA)はラテンシー周期の最終サイクルまで要求されず、サイクル $PC=i+4$ で発生させることができる。

このような手段は理論的には既知であるが、実際には、既に既知の例外事態処理機構に対して複雑となることが知られている。その理由は、問い「この動作は実行前又は実行後に完了したか？」に対する明確な答えがもはや存在しないからである。この複雑さは、例えば、1990年1月のCOMPUTERの'The CYDRA-5 Departmental Supercomputer'でRau等によって言及されている。特に、25頁の項目開始から26頁に続く'Clearly, handling an exception...'参照。したがって、これら考察により、設計者は制御パイプライン配置を削除できないようにし、したがって回路寸法を減少しないようにする。

それに対して、ここで説明する装置のガード機構により、この問いに対する明瞭な答えを許容することができ、したがって設計者は制御パイプライン配置を削除することができる。例外事態が発生すると、例外事態が検出されるとともにガード強制信号GFがアクティブとなる前にベンディング動作に対するガード選択フィールドを発生させる場合及びその場合のみ完了した各ベンディング動作を考察する必要がある。この場合、他のフィールドを発生させるときは、制御パイプライン配置を除去するのを自由に決定する装置及びそのスケジューリングコンパイラの設計者にとって重要である。

ガード選択フィールドの発生のタイミングは、相違する各タイプの動作の詳細な実行に関して相違するので、処理装置の設計者は、再生モードと記録モードとの間の切替を各機能ユニットに対して適切に時間決めするように気をつける必要がある。

当業者は、上記記録／再生機構回路及びその上記及び他の変形を、ここで説明したVLIW処理装置以外の広範囲のデータ処理システムの例外事態回復に対して使

用しうる方法がわかる。

(36)

特許平9-509515

【図1】

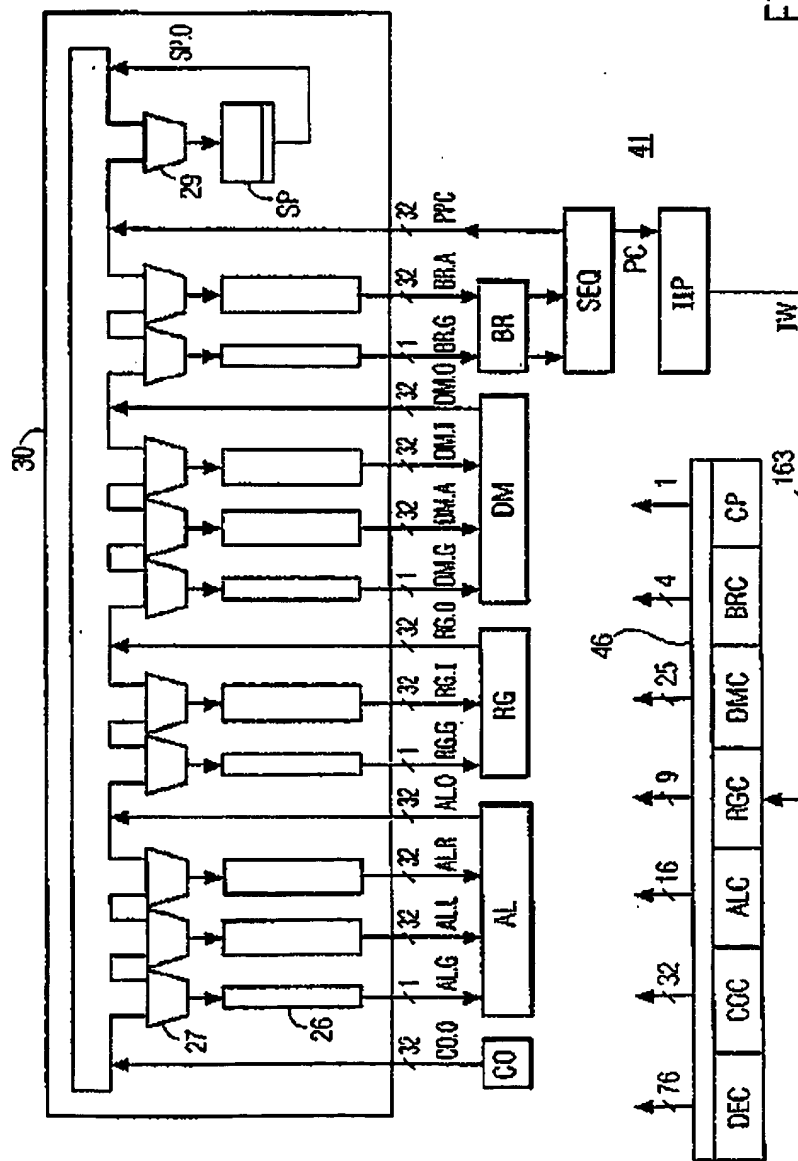


FIG. 1

特許平9-509515

(37)

【図2】

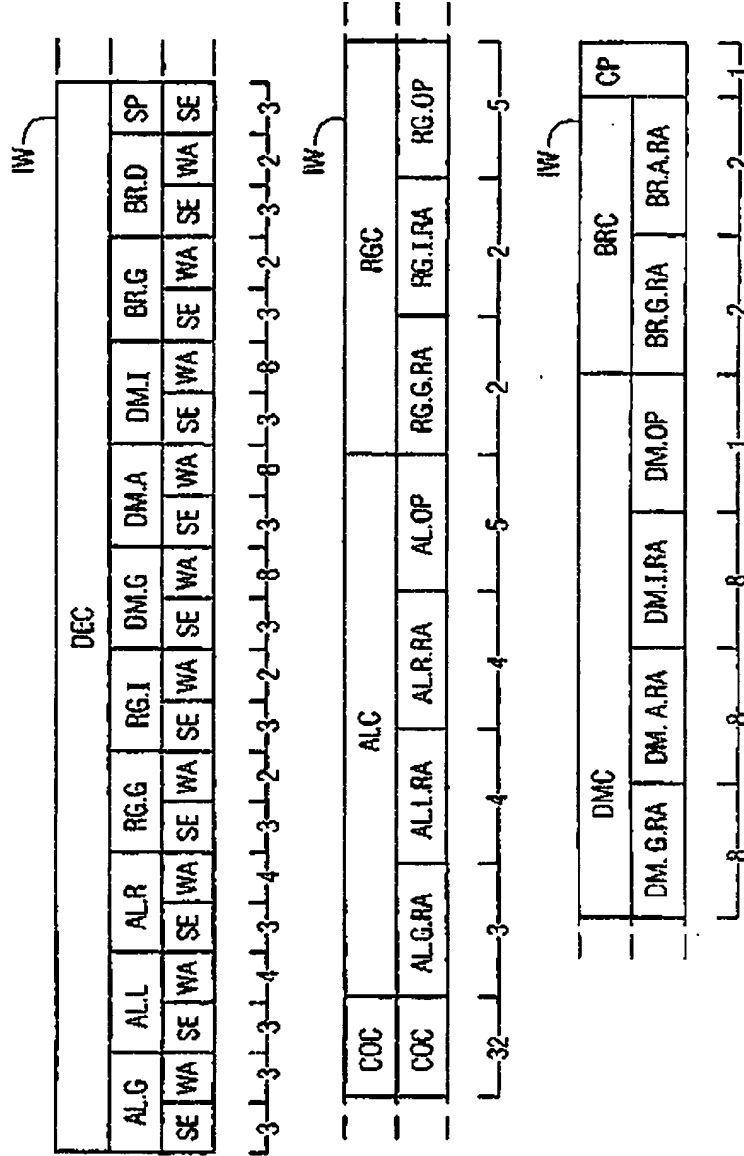


FIG. 2

(38)

特表平9-509515

【図 3】

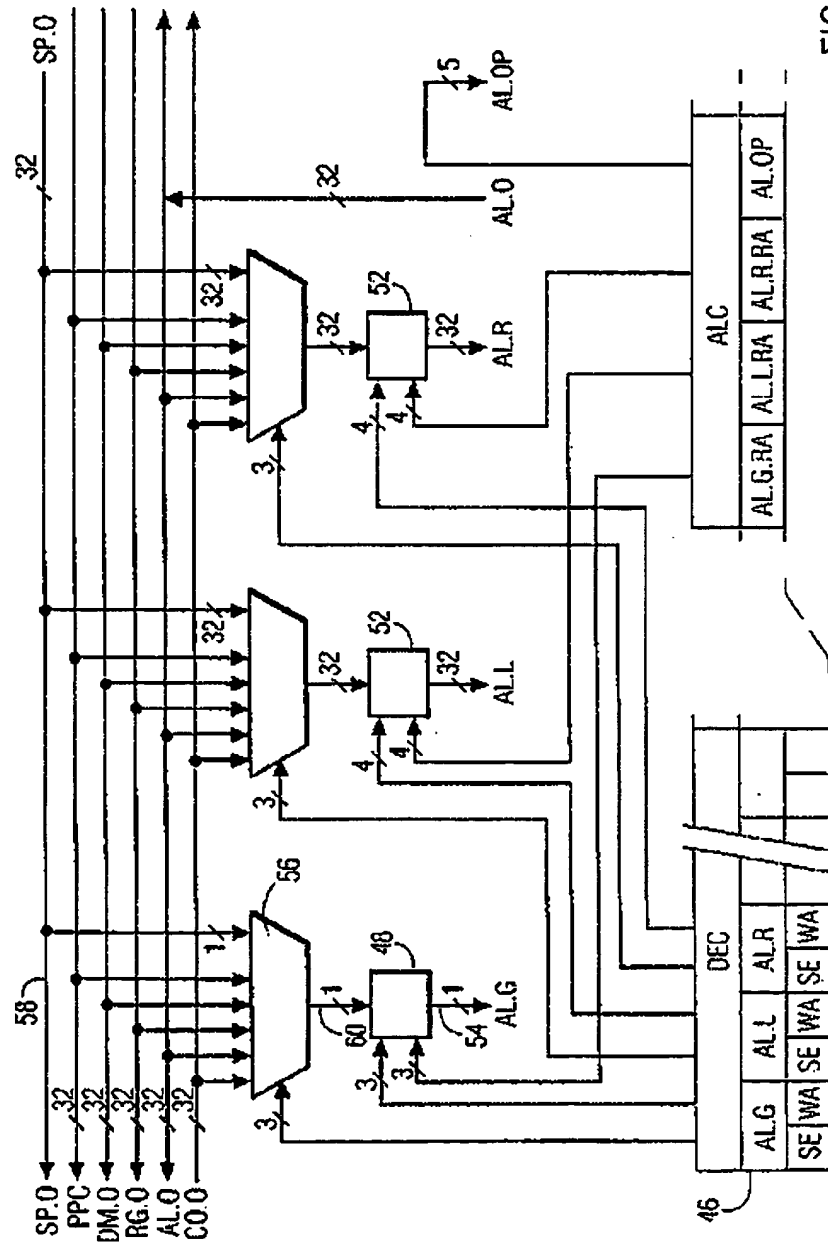


FIG. 3

(39)

特表平9-509515

【図4】

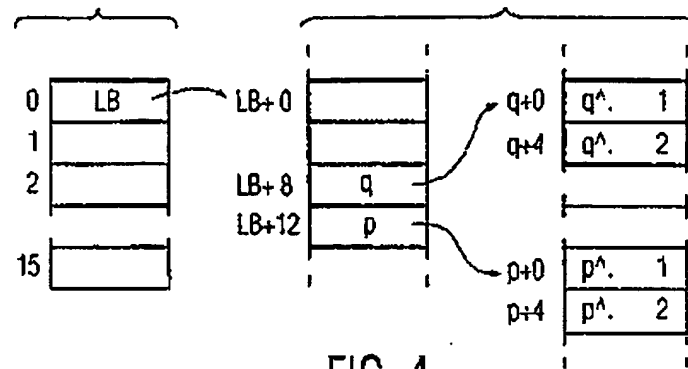


FIG. 4

【図6】

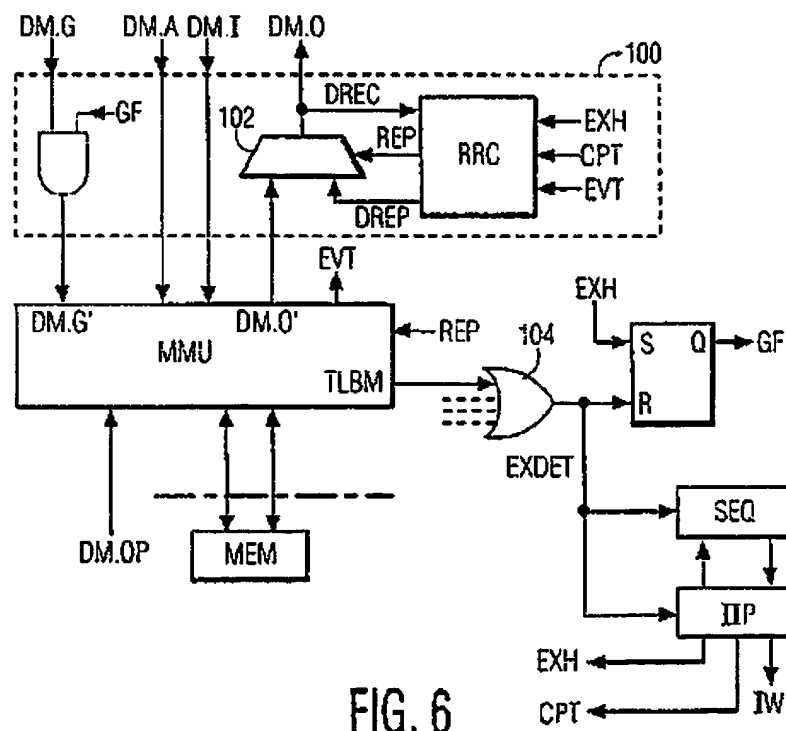


FIG. 6

(40)

特表平9-509515

【図5】

DEC. --- SE

I.

| Cy | COC | ALC | RGC | DMC | BRC | ALG ALL | ALR | RG.G | RG.I | DM.G | DMA | DMI | BR.G | BRA |
|----|-------|-------|-------|-------|-------|---------|-----|------|------|------|-----|-----|------|-----|
| 0 | Op005 | --- | Op001 | --- | --- | --- | CO | --- | --- | --- | --- | --- | --- | --- |
| 1 | Nw015 | Op006 | --- | --- | --- | --- | RG | --- | --- | --- | --- | --- | --- | CO |
| 2 | Op002 | Op003 | --- | Op007 | Nw016 | --- | CO | --- | --- | --- | AL | --- | --- | --- |
| 3 | Op010 | --- | --- | Op004 | --- | --- | CO | --- | --- | --- | AL | --- | --- | --- |
| 4 | --- | Op012 | --- | Op008 | --- | --- | DM | --- | --- | --- | DM | --- | --- | --- |
| 5 | --- | Op011 | --- | Op013 | --- | --- | DM | --- | --- | --- | XX | --- | --- | --- |
| 6 | --- | --- | --- | Op009 | --- | --- | --- | --- | --- | --- | AL | DM | --- | --- |
| 7 | --- | --- | --- | Op014 | --- | --- | --- | --- | --- | --- | --- | DM | --- | --- |

II.

| Cy | COC | ALC | RGC | DMC | BRC | ALG ALL | ALR | RG.G | RG.I | DM.G | DMA | DMI | BR.G | BRA |
|----|-------|-------|-------|-------|-----|---------|-----|------|------|------|-----|-----|------|-----|
| 0 | Op005 | --- | Op001 | --- | --- | --- | CO | --- | --- | --- | --- | --- | --- | --- |
| 1 | Op015 | Op006 | --- | --- | --- | --- | RG | --- | --- | --- | --- | --- | --- | CO |
| 2 | Op002 | Op003 | --- | Op007 | --- | --- | CO | --- | --- | --- | AL | --- | --- | --- |
| 3 | Op010 | --- | --- | Nw016 | --- | --- | CO | --- | --- | --- | AL | --- | --- | --- |
| 4 | --- | Op012 | --- | Op004 | --- | --- | DM | --- | --- | --- | DM | --- | --- | --- |
| 5 | --- | --- | --- | Op008 | --- | --- | --- | --- | --- | --- | AL | --- | --- | --- |
| 6 | --- | Op011 | --- | Op013 | --- | --- | DM | --- | --- | --- | DM | --- | --- | --- |
| 7 | --- | --- | --- | Op009 | --- | --- | --- | --- | --- | --- | AL | DM | --- | --- |
| 8 | --- | --- | --- | Op014 | --- | --- | --- | --- | --- | --- | --- | DM | --- | --- |

III.

| Cy | COC | ALC | RGC | DMC | BRC | ALG ALL | ALR | RG.G | RG.I | DM.G | DMA | DMI | BR.G | BRA | SP |
|----|-------|-------|-------|-------|-------|---------|-----|------|------|------|-----|-----|------|-----|-----|
| 0 | Op005 | --- | Op001 | --- | --- | --- | CO | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Op015 | Op006 | --- | --- | --- | --- | RG | --- | --- | --- | --- | --- | --- | CO | --- |
| 2 | Op002 | Op003 | --- | Op007 | Nw016 | --- | CO | --- | --- | --- | AL | --- | --- | --- | --- |
| 3 | Op010 | --- | --- | Op004 | --- | --- | CO | --- | --- | --- | AL | --- | --- | --- | --- |
| 4 | --- | Op012 | --- | Op008 | --- | --- | DM | --- | --- | --- | DM | --- | --- | --- | --- |
| 5 | --- | --- | --- | Op013 | --- | --- | DM | --- | --- | --- | AL | --- | --- | --- | DM |
| 6 | --- | Op011 | --- | Op009 | --- | --- | --- | --- | --- | --- | SP | DM | --- | --- | --- |
| 7 | --- | --- | --- | Op014 | --- | --- | --- | --- | --- | --- | AL | DM | --- | --- | --- |

FIG. 5

(41)

特表平9-509515

【図7】

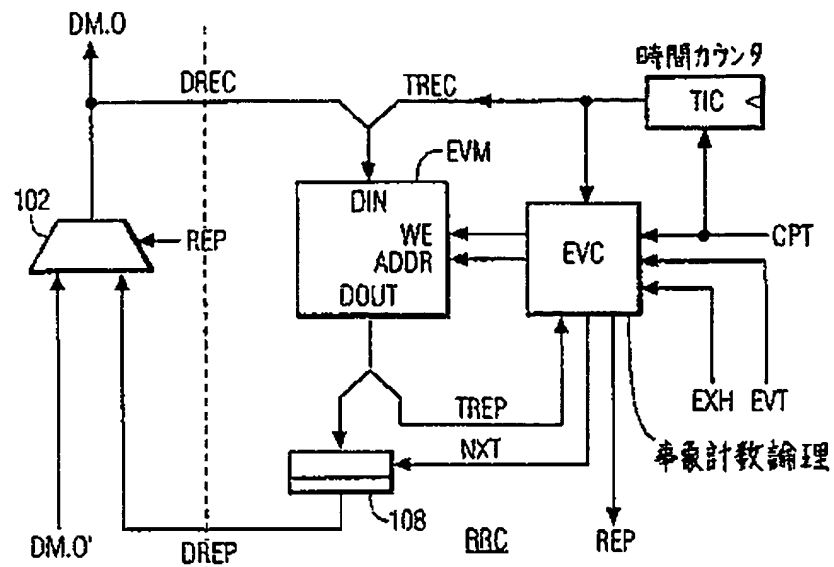


FIG. 7

【図10】

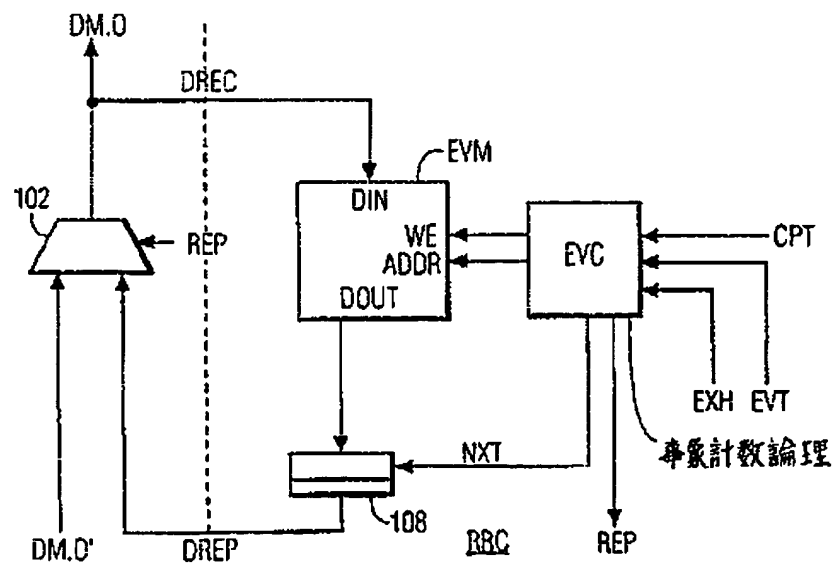


FIG. 10

(42)

特表平9-509515

【図8】

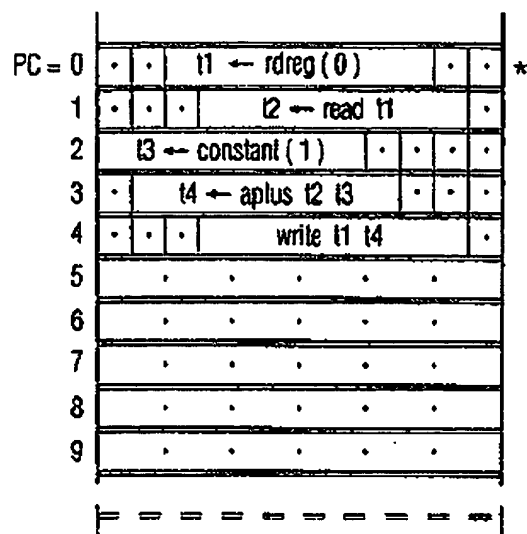


FIG. 8

【図9】

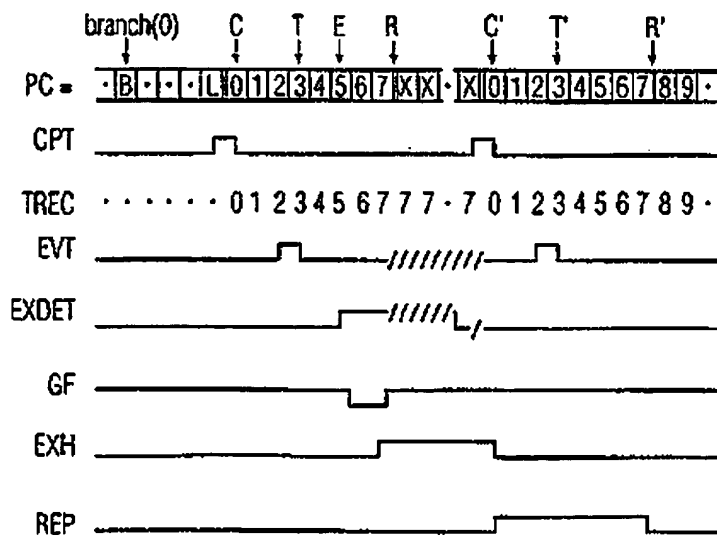


FIG. 9

(43)

特表平9-509515

【国際調査報告】

| INTERNATIONAL SEARCH REPORT | | International application No. PCT/IB 95/01030 |
|---|---|--|
| A. CLASSIFICATION OF SUBJECT MATTER | | |
| IPC6: G06F 11/14 According to International Patent Classification (IPC) or to both national classification and IPC | | |
| B. FIELDS SEARCHED | | |
| Minimum documentation searched (classification system followed by classification symbols) | | |
| IPC6: G06F | | |
| Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched | | |
| Electronic data bases consulted during the international search (name of data base and, where practicable, search terms used) | | |
| EPCDOC | | |
| C. DOCUMENTS CONSIDERED TO BE RELEVANT | | |
| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
| X | EP 0163096 A1 (BBC AKTIENGESELLSCHAFT BROWN, BOVERI & CIE.), 4 December 1985 (04.12.85), abstract | 1-2 |
| Y | abstract | 4 |
| | -- | |
| Y | US 3736566 A (D.W. ANDERSON ET AL), 29 May 1973 (29.05.73), see whole document | 4 |
| | -- | |
| A | EP 0212132 A1 (HEWLETT-PACKARD COMPANY), 4 March 1987 (04.03.87), see whole document | 1-17 |
| | -- | |
| | ----- | |
| <input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex. | | |
| * Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" other document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another claim or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principles or theory underlying the invention "X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is considered with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family | | |
| Date of the actual completion of the international search | | Date of mailing of the international search report |
| 28 June 1996 | | 29-06-1996 |
| Name and mailing address of the ISA: Swedish Patent Office Box 5089, S-102 42 STOCKHOLM Telex No. +46 8 656 02 55 | | Authorized officer Jan Silfverling Telephone No. +46 8 782 25 00 |

Form PCT/ISA/210 (second sheet) (July 1992)

(44)

特表平9-509515

INTERNATIONAL SEARCH REPORT

Information on patent family members

01/04/96

International application No.

PCT/IB 95/01030

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---------------------|----------------------------|---------------------|
| EP-A1- 0163096 | 04/12/85 | DE-A- 3566314 | 22/12/88 |
| | | EP-A, A, A 0254247 | 27/01/88 |
| | | JP-B- 6019722 | 16/03/94 |
| | | JP-A- 60235246 | 21/11/89 |
| | | US-A- 4905196 | 27/02/90 |
| US-A- 3736586 | 29/05/73 | BE-A, A- 787742 | 18/12/72 |
| | | CA-A- 960781 | 07/01/75 |
| | | CH-A- 534925 | 15/03/73 |
| | | DE-A, A, B 2240432 | 01/03/73 |
| | | FR-A, A- 2149996 | 30/03/73 |
| | | GB-A- 1355295 | 05/06/74 |
| | | NL-A- 7211145 | 20/02/73 |
| EP-A1- 0212132 | 04/03/87 | SE-B, C- 380643 | 10/11/75 |
| | | JP-A- 62003347 | 09/01/87 |

Form: PCT/ISA/210 (patent family annex) (July 1992)

JP 1997-509515 AS 2005.4.7

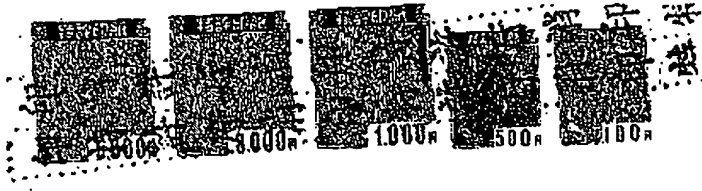
【公報種別】特許法第17条の2の規定による補正の掲載
【部門区分】第6部門第3区分
【発行日】平成17年4月7日(2005.4.7)

【公表番号】特表平9-509515
【公表日】平成9年9月22日(1997.9.22)
【出願番号】特願平8-518521
【国際特許分類第7版】
G 0 6 F 11/14
【F I】
G 0 6 F 11/14 3 1 0 N

【手続補正書】
【提出日】平成16年7月21日(2004.7.21)
【手続補正1】
【補正対象書類名】明細書
【補正対象項目名】補正の内容のとおり
【補正方法】変更
【補正の内容】

(2)

JP 1997-509515 AS 2005.4.7



手 続 補 正 書

(#9600)



平成16年 7月21日

特許庁長官 小川 洋 殿

1 事件の表示

平成 8年 特許願 第518521号



2 補正をする者

名 称 トリメディア、テクノロジーズ インコーポレイテッド

3 代 理 人

住 所 東京都千代田区麹町9丁目2番4号

霞山ビルディング7階 電話(3581)2241 番(代表)

氏 名 (7205) 弁理士 杉 村 豊 作



4 補正により増加する請求項の数 6

5 補正対象書類名 請求の範囲

6 補正対象項目名 請求の範囲



7 補正の内容 別紙の通り

(3)

JP 1997-509515 A5 2005.4.7

1. 請求の範囲を下記の通りに補正する。

「 請 求 の 範 囲

1. プログラムを形成する命令のシーケンスによって制御されるデータ処理システムを作動させるに当たり、

(a) 前記プログラムのチェックポイントの数を規定するとともに前記プログラムを実行し、この実行が前記システム内のデータ転送を伴い、規定されたマシン状態に依存する第1タイプのデータ転送を含むステップと、

(b) このステップ(a)の実行中生じる例外事態に応答して前記プログラムの実行を中断するステップと、

(c) 前記中断の解決後最近のチェックポイントから前記プログラムを再実行するステップとを具え、

前記ステップ(a)は、前記最近チェックポイントから前記第1タイプの各データ転送に関連する記録イベントデータを含み、前記ステップ(c)は、実際のデータ転送の代わりに、前記再実行に同期した、記録されたデータの再生を含み、

前記処理システムが、各動作を並列の実行するよう各命令の各フィールドによって制御される複数の機能ユニットを有し、適切な命令の発生にかかわらず規定マシン状態に影響を及ぼす動作を条件に応じて停止するガード信号を、前記機能ユニットの少なくとも一部に供給し、例外の検出に応答して前記ガード信号を無条件に有効にすることによって例外の検出を禁止する前に、動作を完了しないように構成したことを特徴とするデータ処理システム作動方法。

2. 前記ステップ(c)は、前記実行が前記ステップ(b)で中断された点に前記再実行が到達すると、前記ステップ(a)に応じて前記第1タイプのデータ転送に関連するイベントデータを記録する間の前記プログラムの通常の実行の再開を含むことを特徴とする請求の範囲1記載のデータ処理システム作動方法。

3. 前記ステップ(b)の実行の中断の結果、前記システムの制御を、前記プログラムから例外事態処理ルーチンに転送し、前記制御の転送を遅延して、前記例外事態前に発生した命令により、それらの実行を完了できるようにすることを特徴とする請求の範囲1記載のデータ処理システム作動方法。

4. 前記例外事態処理ルーチンへの制御の転送前の前記遅延中に開始した動作を、

(4)

JP 1997-509515 AS 2005.4.7

前記規定したマシン状態に任意の影響を有しないように禁止することを特徴とする請求の範囲 3 記載のデータ処理システム作動方法。

5. 前記規定されたマシン状態は、前記システムのデータメモリの内容を含み、前記第 1 タイプのデータ転送は、前記データメモリの位置からの値の読出し及びその値を前記システムの処理素子に転送することを含むことを特徴とする請求の範囲 1 記載のデータ処理システム作動方法。

6. 前記規定されたマシン状態を、前記処理システムの全マシン状態のサブセットとし、前記チェックポイントを、前記規定されたマシン状態のみが前記プログラムの次の命令に関連する場合の前記プログラムの点に規定することを特徴とする請求の範囲 1 記載のデータ処理システム作動方法。

7. プログラムを形成する命令のシーケンスによって制御されるデータ処理システムを作動させるに当たり、

(a) 前記プログラムのチェックポイントの数を規定するとともに前記プログラムを実行し、この実行が前記システム内のデータ転送を伴い、規定されたマシン状態に依存する第 1 タイプのデータ転送を含むステップと、

(b) このステップ (a) の実行中生じる例外事態にตอบสนองして前記プログラムの実行を中断するステップと、

(c) 前記中断の解決後最近のチェックポイントから前記プログラムを再実行するステップとを具え、

前記ステップ (a) は、前記最近チェックポイントから前記第 1 タイプの各データ転送に関連する記録イベントデータを含み、前記ステップ (c) は、実際のデータ転送の代わりに、前記再実行に同期した、記録されたデータの再生を含み、

与えられた動作を実行するために、一つより多くの命令のフィールドによって前記処理システムが制御されて、制御パイプラインを減少し、規定マシン状態に影響を及ぼす所定の動作を条件に応じて防止する少なくとも一つのガード信号をシステムに供給し、前記フィールドが、複数の利用できるガード信号から所定の動作のガード信号を選択するガード選択フィールドを有し、他の任意のフィールドとは異なるガード選択フィールドが割込み前に発されたか否かに応じた例外によって生じた割込み前後に生じるような動作を規定することを特徴とするデー

(5)

JP 1997-509515 A5 2005.4.7

ク処理システム作動方法。

8. 前記処理装置が、前記処理システムが、各動作を並列の実行するよう各命令の各フィールドによって制御される複数の機能ユニットを有することを特徴とする請求の範囲7記載のデータ処理システム作動方法。

9. データ処理システムであって、

このシステムによる実行に際しプログラムを形成する命令のシーケンスを発生させる手段と、

前記プログラム中のチェックポイントを識別する手段と、

前記プログラムを実行する間第1タイプのデータ転送を行い、このようなデータ転送の各々が、前記システムの規定されたマシン状態に依存する第1データ経路と、

例外事態にตอบสนองして前記プログラムの実行を中断する手段と、

前記例外事態の解決後の最近チェックポイントからプログラムを再実行する手段とを具えるデータ処理システムであって、

前記システムは、

前記データ経路に接続して、前記最近チェックポイントから前記第1タイプのデータ転送の各々を記録する手段と、

前記再実行中前記データ経路を中断するとともに、実際のデータ転送の代わりに、前記再実行に同期して、記録されたデータ転送を再生する手段と、

適切な命令の発生にかかわらず規定マシン状態に影響を及ぼす動作を条件に応じて停止するガード信号が少なくとも一部に供給される複数の機能ユニットと、

例外の検出にตอบสนองして前記ガード信号を無条件に有効にすることによって、例外の検出を禁止する前に完了しない動作を禁止する手段とを更に具えることを特徴とするデータ処理システム。

10. 前記データ経路を復元するとともに、前記実行が以前に中断された点に前記再実行が到達すると前記再生手段を再起動する手段も含むことを特徴とする請求の範囲9記載のデータ処理システム。

11. 前記例外事態にตอบสนองして、前記システムの制御を前記プログラムから例外事態処理ルーチンに転送し、この制御の転送を遅延させて、前記実行前に開始した

(6)

JP 1997-509515 A5 2005.4.7

動作によりそれらの実行を完了させるようにする手段も含むことを特徴とする請求の範囲 9 記載のデータ処理システム。

12. 前記例外事態処理ルーチンへの制御の転送前の遅延中に開始された動作を禁止して、このような動作が前記規定されたマシン状態に影響を及ぼさないようにする手段も含むことを特徴とする請求の範囲 11 記載のデータ処理システム。

13. 前記規定されたマシン状態は、前記システムのデータメモリの内容を含み、前記データ経路は、前記データメモリの読出しポートを前記システムの処理素子に接続することを特徴とする請求の範囲 9 記載のデータ処理システム。

14. 前記データメモリは仮想記憶システムを具えることを特徴とする請求の範囲 13 記載のデータ処理システム。

15. 命令の再実行中、前記データメモリに対する読出し動作及び書込み動作を抑制するようにしたことを特徴とする請求の範囲 13 記載のデータ処理システム。

16. 前記規定マシン状態を前記処理システムの全マシン状態のサブセットとしたことを特徴とする請求の範囲 9 記載のデータ処理システム。

17. 前記複数の機能ユニットが、各動作を並列の実行するよう各命令の各フィールドによって制御されることを特徴とする請求の範囲 9 記載のデータ処理システム。

18. データ処理システムであって、

このシステムによる実行に際しプログラムを形成する命令のシーケンスを発生させる手段と、

前記プログラム中のチェックポイントを識別する手段と、

前記プログラムを実行する間第 1 タイプのデータ転送を行い、このようなデータ転送の各々が、前記システムの規定されたマシン状態に依存する第 1 データ経路と、

例外事態に応答して前記プログラムの実行を中断する手段と、

前記例外事態の解決後の最近チェックポイントからプログラムを再実行する手段とを具えるデータ処理システムであって、

前記システムは、

前記データ経路に接続して、前記最近チェックポイントから前記第 1 タイプのデータ転送の各々を記録する手段と、

(7)

JP 1997-509515 AS 2005.4.7

前記再実行中前記データ経路を中断するとともに、実際のデータ転送の代わりに、前記再実行に同期して、記録されたデータ転送を再生する手段とを具え、

与えられた動作を実行するために、一つより多くの命令のフィールドによって前記処理システムが制御されて、制御パイプラインを減少し、規定マシン状態に影響を及ぼす所定の動作を条件に応じて防止する少なくとも一つのガード信号をシステムに供給し、前記フィールドが、複数の利用できるガード信号から所定の動作のガード信号を選択するガード選択フィールドを有し、他の任意のフィールドとは異なるガード選択フィールドが割込み前に発されたか否かに応じた例外によって生じた割込み前後に生じるような動作を規定することを特徴とするデータ処理システム。

19. データを記憶する記憶手段と、

データを処理する処理手段と、

前記記憶手段から処理手段へのデータ転送を行うデータ経路と、

プログラムを形成する命令のシーケンスに従って前記記憶手段及び処理手段の動作を制御するシーケンス手段とを具えるデータ処理装置において、

前記データ経路に接続され、(i) 転送されたデータを記録する記録モードと、

(ii) その記録モードで以前に記録されたデータを前記処理手段に供給する間に前記データ経路を中断する再生モードを有する記録／再生手段を更に具え、

与えられた動作を実行するために、一つより多くの命令のフィールドによって前記処理システムが制御されて、制御パイプラインを減少し、規定マシン状態に影響を及ぼす所定の動作を条件に応じて防止する少なくとも一つのガード信号をシステムに供給し、前記フィールドが、複数の利用できるガード信号から所定の動作のガード信号を選択するガード選択フィールドを有し、他の任意のフィールドとは異なるガード選択フィールドが割込み前に発されたか否かに応じた例外によって生じた割込み前後に生じるような動作を規定することを特徴とするデータ処理システム。

20. 前記記録／再生手段は、

前記記憶手段に接続した第1入力部及び前記処理手段に接続した出力部を有することにより前記データ経路に接続したマルチプレクサと、

(8)

JP 1997-509515 AS 2005.4.7

前記データ経路に接続したデータ入力部、前記マルチプレクサの第2入力部に接続したデータ出力部及びアドレス入力部を有する事象メモリ回路と、

(i)記録モード中、前記マルチプレクサにより前記記憶手段を前記処理手段に接続するとともに、前記事象メモリに、前記記憶手段から前記処理手段に転送した各値を記憶させ、(ii)再生モード中、前記マルチプレクサにより前記事象メモリのデータ出力部を前記処理手段に接続するとともに、前記事象メモリにより、前記記録モードで以前に記憶された値を再生する制御回路とを具えることを特徴とする請求の範囲19記載のデータ処理システム。

21. 前記データメモリは仮想記憶システムを具えることを特徴とする請求の範囲19記載のデータ処理システム。

22. 前記処理手段は、プログラム命令の各フィールドの制御の下で互いに同期しながら作動するよう構成された複数の機能ユニットを具えることを特徴とする請求の範囲19記載のデータ処理システム。

23. 前記データメモリは仮想記憶システムを具えることを特徴とする請求の範囲22記載のデータ処理システム。」